

UNIVERZA V LJUBLJANI
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Primož Rome

**ORODJE ZA OZNAČEVANJE
IN ANALIZO ČLOVEŠKIH
OBRAZOV NA SLIKAH**

Diplomska naloga na
visokošolskem študiju

Mentor: prof. dr. Franc Solina

Asistent: dr. Peter Peer

Ljubljana, april 2006

Zahvala

Najlepše se zahvaljujem svojemu mentorju prof. dr. Francu Solini pri izbiri in usmerjanju teme za diplomsko nalogo, raziskovalcu dr. Petru Peeru za pomoč pri načrtovanju razvoja, spremljanju napredka pri izdelavi diplomske naloge in nasploh veliko časa, porabljenega za koristne nasvete pri izdelavi naloge. Zahvaljujem se tudi mlademu raziskovalcu mag. Borutu Batagelju za priskrbljeno literaturo in pomoč pri testiranju orodja.

Zahvala tudi sodelavcem v podjetju Hermes Softlab d.d., s katerimi sem prišel do znanja in izkušenj s področja razvoja programske opreme, ki sem jih uporabil v diplomski nalogi. Posebna zahvala gre sodelavcu Mihi Vidmarju, ki mi je pomagal pri specifičnih problemih implementacije.

Posebno bi se rad zahvalil še moji družini, ki mi je omogočila študij in mi ves čas študija stala ob strani; posebej bi se zahvalil svojemu očetu, ki mi je odprl vrata v svet računalništva.

Kazalo

| | |
|---|----------|
| POVZETEK | 1 |
| 1 UVOD | 2 |
| 1.1 PREPOZNAVA ČLOVEŠKIH OBRAZOV NA SLIKAH | 3 |
| 1.1.1 Sistem za iskanje in razpoznavanje obrazov | 4 |
| 1.2 PODATKOVNE BAZE | 5 |
| 1.2.1 Zasnova in zajem podatkovne baze | 6 |
| 1.2.2 Zajem podatkovne baze z označeno pozo | 7 |
| 1.2.2.1 Uporaba navideznih oznak | 8 |
| 1.2.2.2 Uporaba senzorjev in kalibriranih kamer | 8 |
| 1.2.3 Javno dostopne podatkovne baze obrazov | 9 |
| 1.3 DELA S PODOBNO TEMATIKO | 13 |
| 1.3.1 Orodje LabelME | 13 |
| 1.3.2 Ostala orodja | 15 |
| 1.3.3 Ugotovitev | 15 |
| 2 ORODJE ZA OZNAČEVANJE IN ANALIZO ČLOVEŠKI OBRAZOV NA SLIKAH...17 | |
| 2.1 CILJI IN ZAHTEVE | 17 |
| 2.2 NAČRTOVANJE ORODJA | 20 |
| 2.2.1 Analiza uporabniškega modela | 21 |
| 2.2.1.1 Uporabnik označevanja | 21 |
| 2.2.1.2 Administrator sistema | 22 |
| 2.2.1.3 Uporabnik podatkovne baze | 23 |
| 2.2.1.4 Uporabnik analize | 24 |
| 2.2.2 Sklep | 25 |
| 2.3 ARHITEKTURA ORODJA FAAT | 25 |
| 2.3.1 Avtentikacija uporabnikov | 27 |
| 2.3.1.1 Funkcionalnost in uporabnost | 28 |
| 2.3.2 Aplikacija za označevanje slik | 29 |
| 2.3.2.1 Funkcionalnost in uporabnost | 30 |
| 2.3.3 Aplikacija za prenos baze slik | 32 |
| 2.3.3.1 Funkcionalnost in uporabnost | 32 |
| 2.3.4 Aplikacija za analizo rezultatov | 34 |
| 2.3.5 Funkcionalnost in uporabnost | 34 |
| 2.4 FDRML | 34 |
| 2.4.1 Uvod | 34 |
| 2.4.2 Struktura in zapis podatkov | 35 |
| 2.5 REZULTATI ALGORITMOV | 37 |
| 2.5.1 Struktura podatkov | 37 |
| 2.6 NAMESTITEV ORODJA FAAT | 39 |
| 2.6.1 Strežnik | 39 |
| 2.6.2 Podatkovna baza | 39 |
| 2.6.3 Namestitev | 40 |

| | | |
|----------|--|-----------|
| 3 | SKLEPNE UGOTOVITVE | 42 |
| 4 | PRILOGE | 44 |
| | A. PRIMER KREIRANEGA XML DOKUMENTA | 44 |
| | B. XML SHEMA FDRML JEZIKA | 47 |
| | C. DATOTEKA Z REZULTATI ALGORITMA | 54 |
| | SPISEK SLIK..... | 56 |
| | SPISEK TABEL | 57 |
| 5 | LITERATURA IN VIRI | 58 |
| | IZJAVA O AVTORSTVU | 59 |

Seznam ključnih besed in kratic

Računalniški vid

Podatkovna baza obrazov

Človeški obraz

Značilke obraza

Označitev

Analiza

Spletni strežnik

Spletni odjemalec

Face Annotation and Analysis Tool – FAAT

Face Detection Results Markup Language – FDRML

Extensible Markup Language – XML

Asynchronous JavaScript And XML – AJAX

Povzetek

Diplomska naloga se navezuje na raziskovalno področje računalniškega vida. Obsega pregled problematike izgradnje podatkovne baze človeških obrazov in analizo rezultatov algoritmov za prepoznavanje in detekcijo človeških obrazov. Začrtano je orodje za označevanje in analizo človeških obrazov na slikah. Orodje je bilo narejeno z uporabo sodobnih tehnologij, katere pripomorejo k izboljšanju uporabniške izkušnje. V okviru orodja je implementiranih več aplikacij: Aplikacija za označevanje slik, aplikacija za administracijo sistema, aplikacija za pridobitev podatkovne baze obrazov in aplikacija za analizo rezultatov algoritmov.

V uvodnem delu diplomske naloge sem podal problematiko izgradnje podatkovnih baz obrazov, poudaril njihovo pomembnost pri razvoju in evaluiranju algoritmov ter predstavil nekaj obstoječih podatkovnih baz. Predstavil sem nekaj sorodnih obstoječih rešitev, poudaril njihove prednosti in pomanjkljivosti ter potegnil vsporednice med njimi in načrtanim orodjem.

V glavnem delu sledi opis postopka načrtovanja in izgradnje aplikacij orodja. Podane so slike shem, ki so bile kasneje uporabljene v fazi implementacije. Predstavljene so nekatere tehnike načrtovanja in izgradnje orodja, temu pa sledijo opisi posameznih aplikacij in modulov. Predstavljen je jezik za opis rezultatov algoritmov, katerega smo razvili v okviru razvoja orodja.

Poglavje tri zajema sklepne ugotovitve diplomske naloge ter predstavi slabosti in priporočila možne izboljšave orodja.

1 □ Uvod

Živimo v času, ko potreba po vizualni predstavitvi podatkov raste iz dneva v dan. Raziskave na področju vida, od grških filozofov, prek gestaltске šole psihologije na začetku 20. stoletja do današnjih dni, kažejo na to, da je proces ustvarjanja mentalne predstave o prizoru, ki ga človek opazuje, lahko pojasnjen s spletom različnih mehanizmov. Dvodimenzionalna slika je le eden izmed osnovnih faktorjev mentalne predstavitve prizora, vendar med vsemi faktorji podaja veliko, če ne kar največ informacij o opazovanem prizoru. Pri določenih problemih računalniškega vida je tako praktično nesmiselno delati s kompleksnejšimi modeli, če nam že enostavni ponujajo dovolj veliko količino informacij.

Računalniški vid je disciplina oz. tehnologija, s katero skušamo nadomestiti človeški vid. Eden njenih ciljev je iskanje objektov na dvodimenzionalnih slikah. Tako kot človek neznan prizor preučuje na način, da ga "razbija" na manjše dele in primerja z znanimi vzorci iz spomina, tudi postopek iskanja objektov segmentira sliko.

Zaradi vse manjših omejitev pri shranjevanju in prenosu vizualnih vsebin, vse bolj razširjenih in obsežnih slikovnih podatkovnih baz in »živih« video informacij je njihovo inteligentno oziroma avtomatično preiskovanje izredno pomembno. Ljudje, to so človeški obrazi, so eden od najpogostejših in zelo specifičnih objektov, ki jih želimo poiskati na slikah. O pomembnosti problematike pričajo številne raziskave, ki jih na tem področju izvajajo raziskovalne skupine po svetu.

V svetu računalništva nobeno področje ni majhno v smislu, da ga lahko človek povsem obvlada. Na vsako področje namreč vpliva toliko drugih področij (računalniških in ostalih), da enostavno ne moremo potegniti razločne meje med posameznimi področji. Ideja z nekega področja je nemalokrat uporabna tudi drugje, slediti tempu, ki ga narekuje takšno življenje, pa je praktično nemogoče in nesmiselno! Zato so strokovnjaki v današnjem svetu največkrat "ozko usmerjeni", kar ni nujno dobro, je pa potrebno za uspešno prilagajanje okolju, v katerem živimo!

Tako je seveda tudi pri sistemih za avtomatsko iskanje človeških obrazov na slikah in njihovimi izvedenkami oziroma nadgradnjami! Veliko avtorjev se že več let ukvarja le z določenim podproblemom takšnega sistema, takšnih podproblemov pa

je spet mnogo. Rezultati, ki jih dosegajo posamezni algoritmi (sistemi), so primerljivi med seboj, če upoštevamo njihovo kompleksnost in omejitve. Na prvi pogled bi lahko rekli, da vsi algoritmi dosegajo podobne rezultate, noben od njih pa še vedno ni “popoln”. Večina razvitih algoritmov ima vsaj enega od dveh najpogostejših problemov:

- prevelika računska (časovna, prostorska) kompleksnost in/ali
- premajhna učinkovitost.

Na osnovi teh dveh problemov je bila načrtovana pot, katere rezultat je pričujoče delo. Delo obsega pregled sistemov in podatkovnih baz za testiranje in analizo rezultatov algoritmov za avtomatsko iskanje človeških obrazov na slikah. V delu je predstavljena rešitev za ročno označevanje človeških obrazov in značilk na slikah ter primerjava ročnih označb z rezultati algoritmov.

1.1 Prepoznavanje človeških obrazov na slikah

Pod prepoznavo človeških obrazov na slikah razumemo sistematično-avtomatiziran proces, ki na območju digitalno zajete slike prepozna enega ali več človeških obrazov. Za prepoznavo človeških obrazov je potrebno razviti temu namenjen sistem in znotraj njega implementirati algoritme, ki na osnovi zajetih slik uspešno prepoznavajo človeške obraze, če so seveda prisotni. Uspešnost samih algoritmov lahko zelo varira in je odvisna od mnogih spremenljivk. Da bi dosegli čim večjo uspešnost in fleksibilnost implementiranih algoritmov, potrebujemo ustrezno okolje, ki nam omogoča fleksibilno testiranje in analizo rezultatov posameznih algoritmov.

Na tem mestu je potrebno izpostaviti dejstvo, da je avtomatsko iskanje obrazov, kot tudi večina drugih avtomatskih iskanj različnih objektov, zahtevna naloga; predvsem zaradi pomembnih variacij vzorcev, ki jih težko parametriziramo analitično.

Namen algoritmov za avtomatsko iskanje človeških obrazov na slikah je očiten: njihov osnovni cilj je segmentirati sliko na regije, ki vsebujejo obraz ali njegove dele, in na regije, ki jih lahko v nadaljnji obdelavi slike zanemarimo, saj ne predstavljajo obraza ali njegovega dela. Izpostavimo nekaj aplikacij, ki uporabljajo takšne algoritme:

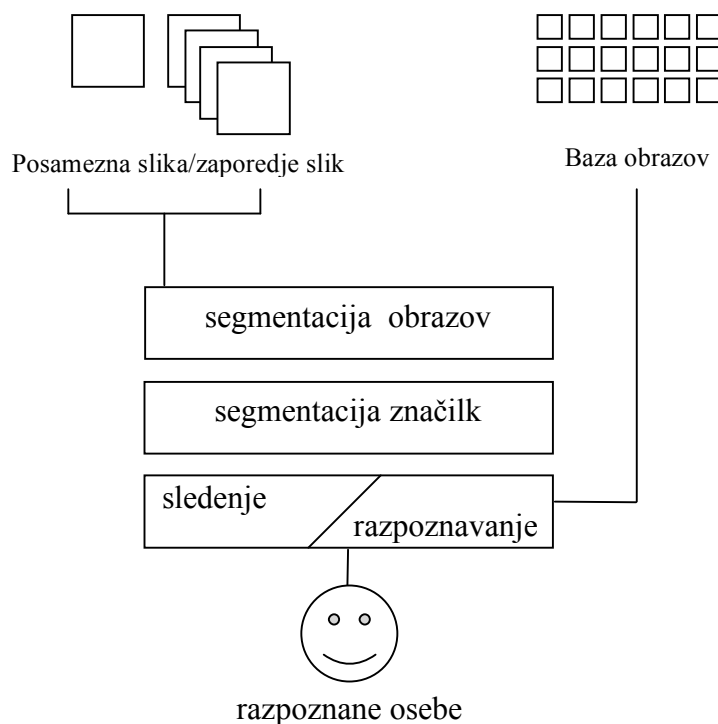
- aplikacije za nadzor in varnost, ki predstavljajo najpomembnejšo skupino aplikacij,
- aplikacije za video konferenčni sistem,
- aplikacije grafične animacije mimike uporabnika,
- aplikacije oddaljenega vodenja kamere,
- umetniške inštalacije,
- drugo.

1.1.1 Sistem za iskanje in razpoznavanje obrazov

V grobem lahko rečemo, da nek sistem za iskanje in razpoznavanje obrazov sestavljajo naslednji deli (slika 1):

- zaznavanje gibanja,
- iskanje obraza (obrazov),
- sledenje gibanju obraza,
- izločanje značilnosti obraza in
- prepoznavanje obraza.

Takšen sistem v realnem svetu predstavlja aplikacijo nadzora in varnosti, algoritme takšnega sistema pa lahko uporabljamo še marsikje (npr. pri nadzoru kvalitete izdelkov).



Slika 1: Algoritem sistema za iskanje in razpoznavanje obrazov.

Vsak del je največkrat spet (kompleksna) zbirka algoritmov, ki realizirajo določene zahteve. Velikokrat z algoritmi računalniškega vida v takšnih sistemih sodelujejo tudi različni pristopi k strojnemu učenju (največkrat se uporabljajo nevronske mreže). Vendar pa je več kot jasno, da strojno učenje ni vedno na mestu, saj lahko realiziramo določene zahteve bolj učinkovito brez njega!

Povsem logično je, da so algoritmi za prepoznavo obrazov namenjeni prepoznavanju obrazov. Univerzalni algoritmi, to so takšni, ki bi lahko delovali v kakršnemkoli okolju in kadarkoli, ne obstajajo. Zaradi kompleksne strukture je prikaz obraza odvisen od mnogih faktorjev, med njimi naslednjimi: identiteta, položaj obraza, osvetlitev, izraz, starost, narodnost, itd. Takoj je jasno, da so algoritmi optimizirani za prepoznavanje neke ciljne skupine ljudi oz. populacije. Razvoj algoritmov, robustnih na vse te variacije, zahteva podatkovno bazo zadostne velikosti in skrbno kontroliranih variacij zgoraj omenjenih faktorjev. V naslednjih podpoglavjih so predstavljeni pristopi k izgradnji takšnih podatkovnih baz in sama predstavitev javno dostopnih baz, ki jih lahko uporabimo z namenom testiranja in evaluacije algoritmov.

1.2 Podatkovne baze

Razvoj sistema za prepoznavo obrazov zahteva izgradnjo podatkovne baze posameznih slik oz. zaporedja slik. Sestava takšne podatkovne baze je v veliki meri odvisna od namena in ciljnih funkcionalnosti sistema za razpoznavo obrazov. Gledano široko se lahko vsaka baza človeških obrazov uporabi v namen:

- a) razvoja in evaluacije algoritmov za prepoznavo človeških obrazov ali
- b) vnos slik obrazov za uporabo v aplikacijah.

Algoritem za učenje in prepoznavo obrazov potrebuje podatkovno bazo slik, na katerih so predstavniki ciljnih variacij, za katere pričakujemo, da jih algoritem prepozna. Razvoj takšnih, predvsem robustnih in konsistentnih algoritmov je zelo težak. Zaradi tega je potrebno variacije pazljivo kontrolirati in, če je možnost variacije, množiti. Podatkovna baza slik, uporabljena za evaluiranje kateregakoli sistema za prepoznavo obrazov, mora biti formirana tako, da vsebuje variacije obrazov, čimbolj podobnih tistim v fazi razvoja samega sistema.

1.2.1 Zasnova in zajem podatkovne baze

Zajem podatkovne baze zahteva pazljivo preučevanje tega, kako kontrolirati več spremenljivk, ki ciljno definirajo možne pojavitvene variacije na slikah z obrazi. Meja, do kamor bodo te spremenljivke še sistematično kontrolirane, bo neposredno odvisna od ciljev študije, ki bo izvedena z uporabo te podatkovne baze. Izvorne karakteristike, ki vplivajo na obrazne variacije na slikah, na grobo delimo v dve skupini:

- notranje karakteristike (variacije) subjekta (*intrinsic variations*)
- zunanje karakteristike (variacije), ki vplivajo na subjekt (*extrinsic variations*).

Notranje karakteristike izvirajo iz fizikalne narave ljudi in njihovih obrazov. Ob sestavljanju podatkovne baze obrazov lahko premislimo o naslednjih spremenljivkah notranjih karakteristik:

- starost,
- spol,
- narodnost.

Podatkovna baza naj bi bila zastavljena ožje in naj bi namesto širše ali celotne populacije vsebovala samo variacije subjektov, na katere cilja zastavljena aplikacija. Na primer, sistem za kontroliranje dostopa do Tokijske banke naj ne bi bil razvit z uporabo podatkovne baze obrazov z veliko kavkaških in otroških primerkov.

Da bi razvili sistem, ki je dolgoročno odporen na staranje ciljne populacije, bomo primorani vpoklicati subjekte na več zajemov slik v različnih mesečnih ali letnih intervalih.

Na točki, ko nam je znano število in narava subjektov, smo pripravljeni za zajem podatkov (slik). Med zajemom določenega obraza moramo kontrolirati naslednje spremenljivke notranjih karakteristik:

- **Izraz obraza:** Kakšne variacije izrazov na obrazu so zaželjene in kako jih kontroliramo? Kakšna navodila dajemo subjektom med zajemom različnih variacij?
- **Oblačila:** So kakšne omejitve glede oblačil? Na primer, je dovoljena prisotnost kap, šalov, kapuc, ipd.?

- **Okrasitev obraza:** Do kakšne mere so obrazi subjektov okrašeni z dodatki kot so nakit, ličila in očala?
- **Lasje:** Ali naj bodo lasje zakriti, na primer s kopalno kapico?

Zunanje karakteristike so spremenljivke okolja, ki vplivajo na obraze subjektov. Okolje, kjer poteka zajemanje slik, zelo vpliva na sam zajem, prav tako pa tudi naprava, ki služi zajemanju. Pred zajemom slik je potrebno pregledati in preučiti naslednje spremenljivke zunanjih karakteristik:

- **Poravnava:** Obrazne slike so lahko poravnane med zajemom tako, da jih poravnamo glede na obrazno značilko, na primer oko.
- **Ozadje:** Homogeno ozadje je lahko zelo koristno. Skrbno pripravljeno ozadje nam lahko služi za tehniko avtomatske segmentacije.
- **Nastavitve kamere:** Uporabljene leče in oddaljenost kamere od subjekta zelo vplivajo na kvaliteto slike in na stopnjo prisotnosti efektov perspektive. Prav tako je zelo pomemben fokus kamere.
- **Naprava za slikanje:** Najbolj pogosto se uporabljajo monokularne CCD kamere, ki zajemajo 2-dimenzionalno sliko ali sekvenco teh slik. Alternativno lahko uporabimo binokularne ali laserske sisteme za 3-dimenzionalni zajem obrazov.
- **Svetloba:** Okolje, kjer poteka zajemanje, ima lahko različne vire in tipe svetlob. Podrobneje lahko preučimo, kakšen vir svetlobe je za nas najbolj primeren.
- **Poza:** Mehanizmi in natančnost pozicije glave glede na objektiv kamere.
- **Velikost:** Širina in višina obraza na sliki.

Obrazne slike so lahko po zajemu podatkov specifično naravnane in obdelane, tako da tvorijo primerno strukturo za učenje algoritmov. Tega procesa ne moremo popolnoma avtomatizirati, zato se mu je boljše izogniti ali si proces vsaj delno olajšati in slike primerno naravnati že med samim zajemom.

1.2.2 Zajem podatkovne baze z označeno pozo

Trenutno je dostopnih dokaj veliko podatkovnih baz obrazov, kjer so obrazi zajeti le iz sprednjega pogleda, primanjkuje pa podatkov, kjer so obrazi sistematično zajeti pod več koti (neprekinjeno zajeti pod koti od -90° do 90°). Poglavje opisuje zajem takšne podatkovne baze in kontroliranje različnih spremenljivk, opisanih v poglavju 1.2.1.

1.2.2.1 Uporaba navideznih oznak

Preprost način za zajem podatkovnih baz z označeno pozo je uporaba navideznih oznak. Postopek zajema slik je sledeč: Medtem, ko subjekt sedi na vrtljivem stolu v sredini navideznega polkroga, namestimo navidezne oznake na loku (v intervalih po 10°), omenjenega navideznega polkroga. Navidezne oznake namestimo v isti višini, kot so oči subjekta. Kamera je tudi nameščena v isti višini kot navidezne oznake. Pred začetkom zajema slik poskrbimo, da je subjekt (njegov obraz od spredaj) natančno poravnan s centrom kamere. Zajem slik izvršimo tako, da prosimo subjekt, da pogleda v nameščene navidezne oznake. Navidezne oznake so iz pogleda subjekta nameščene v intervalu po 10° v loku do $+90^\circ$ (desni profil) in v loku do -90° (levi profil).

Slabost metode navideznih oznak je v tem, da je označitev poz lahko neenakomerna in slabo porazdeljena. Temu je tako, ker so nekateri subjekti naravno nagnjeni k premikanju oči, namesto k obračanju glave proti postavljenim navideznim oznakam. Prav tako je s to metodo zelo težko kontrolirati in meriti naklon glave subjekta. Kljub slabostim pa ima metoda še vedno veliko prednost v enostavni in poceni namestitvi takšnega zajema. Zajete slike po tej metodi lahko uporabimo v začetne eksperimente in njihovo uporabo v komercialnih aplikacijah. Bolj zahtevna analiza in evaluiranje algoritmov za razpoznavo in detekcijo obrazov, pa že zahteva natančnejšo metodo zajema in označitve slik.

1.2.2.2 Uporaba senzorjev in kalibriranih kamer

Sistem, ki uporablja elektro-magnetni senzor in kalibrirano kamero je bil narejen za zajemanje natančno poravnanih slik obrazov (označenih poz). Rezultat zajema takšnih slik, so natančno in enakomerno poravnani obrazi, zajeti pod različnimi koti. Uporabljen je sledilnik z senzorjem in oddajnikom. Ta omogoča 3D koordinate in orientacijo senzorja, relativno glede na oddajnik. Senzor nato nepremično namestimo, na čeladi podobno stvar, in jo namestimo na glavo subjekta. Senzor sledi gibom glave in spreminja orientacijo. Monokularni sistem za zajemanje slik je kalibriran na 3D koordinatni sistem senzorja in omogoča zajem 3D pozicije obraza relativno glede na senzor.

Za zajem 2D slike obraza moramo pravilno kalibrirati kamero glede na senzor, ki je nameščen na glavi subjekta. Kalibriranje kamere zajema nastavitve množice parametrov, tako spremenljivk notranjih karakteristik (goriščna razdalja in radialno popačenje) kot spremenljivk zunanjih karakteristik (relativna pozicija in orientacija kamere glede na koordinatni sistem senzorja).

Položaj in velikost glave varira od subjekta do subjekta, vendar moramo, neglede na to poskrbeti za enakomeren zajem slik. Zato je potrebno za natančen zajem podatkov poskrbeti tudi za pravilno poravnanje glave subjekta. Prvi korak je določitev 3D koordinat posameznih obraznih značilk relativno glede na senzor. To storimo tako da lociramo oči in sredino zgornje ustnice subjekta s sprednjega pogleda. Ti dve značilki obraza so vedno na istem mestu glede na obraz. Operator na zaslonu začrta dva vodoravno vsporedna kvadrata (s središčem kvadratov v levem in desnem očesu) in navpično črto med njima. Te označbe operaterju pomagajo najti simetrični sprednji pogled na obraz, ki služi zajemu 2D slike obraza pod različnimi koti. Slike so glede na postavljene koordinate na obrazu še pravilno obrezane in nato zajete.

1.2.3 Javno dostopne podatkovne baze obrazov

Zajem visoko kvalitetne podatkovne baze obrazov za namene testiranja in analize algoritmov je dolgotrajen proces, kjer poleg časa potrebujemo še veliko število različnih virov. Dostopnost javnih podatkovnih baz obrazov je zelo pomembna za napredovanje znanosti računalniškega vida. V tem podpoglavju bom naštel in opisal nekaj javno dostopnih podatkovnih baz obrazov, ki so na voljo za odkrivanje in razpoznavo obrazov.

V grobem lahko javno dostopne baze obrazov delimo na tri vrste:

- baze za razpoznavo obrazov,
- baze za odkrivanje obrazov in
- baze za analizo obraznih izrazov.

Podatkovna baza AR

| Št. subjektov | Variacije | Ločljivost slike | Št. slik |
|---------------|--|------------------|----------|
| 116 | Izrazi 4 Osvetljenost 4 Prekrivanje 2 Čas 2 | 768 x 576 | 3288 |

Tabela 1: Statistika podatkovne baze obrazov AR.

Podatkovno bazo so zajeli leta 1998 v centru za računalniški vid v Barceloni, v Španiji. Baza vsebuje slike 116 posameznikov (63 moških in 53 žensk). Zajem in pogoji zajema (nastavitve kamere, svetloba) so bili skrbno kontrolirani za čimbolj

identičen zajem med subjekti. Subjekti so bili posneti dvakrat v dvotedenskem intervalu.

Dostopno na http://rv11.ecn.purdue.edu/aleix/aleix_face_DB.html.

Podatkovna baza BANCA

| Št. subjektov | Variacije | Ločljivost slike | Št. slik |
|---------------|----------------------|------------------|----------|
| 208 | Kvaliteta 2 Čas 2 | 720 x 576 | Neznano |

Tabela 2: Statistika podatkovne baze obrazov BANCA.

Ta podatkovna baza je bila zajeta v okviru evropskega projekta BANCA, katerega cilj je bil razvoj in implementacija varnostnega sistema z naprednim sistemom avtentikacije, identifikacije in dostopnih shem za aplikacije, ki delujejo preko interneta. Podatkovna baza je bila zasnovana za testiranje verifikacije identitete z različnimi napravami za zajem slike in pod različnimi pogoji.

Dostopno na <http://www.ee.surrey.ac.uk/Research/VSSP/banca/>.

Podatkovna baza CAS-PEAL

| Št. subjektov | Variacije | Ločljivost slike | Št. slik |
|---------------|--------------|------------------|----------|
| 1040 | Poza 21 | 360 x 480 | 30.900 |
| 377 | Izraz 6 | | |
| 438 | Dodatki 6 | | |
| 233 | Svetloba 6 | | |
| 297 | Ozadje 9-15 | | |
| 296 | Razdalja 1-2 | | |
| 66 | Čas 2 | | |

Tabela 3: Statistika podatkovne baze obrazov CAS-PEAL.

Baza je bila zajeta na Kitajski akademiji znanosti (Chinese Academy of Sciences), med avgustom leta 2002 in aprilom leta 2003. Baza zajema slike subjektov različnih spolov v sedmih kategorijah: poza, izraz, dodatki, svetloba, ozadje, razdalja in čas. Vsaka podkategorija zajema večje število variacij.

Dostopno na <http://www.jdl.ac.cn/peal/index.html>.

Podatkovna baza CMU Hyperspectral

| Št. subjektov | Variacije | Ločljivost slike | Št. slik |
|---------------|-----------------------|------------------|----------|
| 54 | Svetloba 4 Čas 1-5 | 640 x 480 | Neznano |

Tabela 4: Statistika podatkovne baze obrazov CMU - Hyperspectral.

CMU baza je bila zajeta v okviru programa »HumanID«. Vse slike so bile zajete z uporabo samorazvitega slikovnega senzorja na podlagi akustično-optičnega filtra (Acousto-Optic-Tunable Filter). Zaradi slabe zaznave svetlobe senzorja so med zajemom uporabljali 600W halogenske luči.

Dostopno na http://www.ri.cmu.edu/pubs/pub_4110.html.

Podatkovna baza CMU PIE

| Št. subjektov | Variacije | Ločljivost slike | Št. slik |
|---------------|-----------------------------------|------------------|----------|
| 68 | Poza 13 Svetloba 43 Izraz 3 | 640 x 486 | 41.368 |

Tabela 5: Statistika podatkovne baze obrazov CMU PIE.

Baza je bila zajeta med oktobrom in decembrom 2000. Sistematično zajema veliko število subjektov pod različnimi svetlobnimi pogoji, različnimi pozami in z različnimi izrazi. 68 posameznikov je bilo zajetih v posebno pripravljeni sobi z uporabo 13 sinhroniziranih in visoko kvalitetnih barvnih kamer in 21 bliskavic.

Dostopno na http://www.ri.cmu.edu/projects/project_418.html.

Podatkovna baza FERET

| Št. subjektov | Variacije | Ločljivost slike | Št. slik |
|---------------|---|------------------|----------|
| 1199 | Izraz 2 Svetloba 2 Poza 9-20 Čas 2 | 256 x 384 | 14.051 |

Tabela 6: Statistika podatkovne baze obrazov FERET.

Baza je nastala s sodelovanjem univerze Georga Masona in ameriškega vojaškega laboratorija za raziskave v okviru programa FERET (The Facial Recognition Technology). Slike so bile posnete v petnajstih izmenah med letoma 1993 in 1996.

Dostopno na <http://www.nist.gov/humanid/feret>.

Korejska podatkovna baza obrazov (KFDB)

| Št. subjektov | Variacije | Ločljivost slike | Št. slik |
|---------------|----------------------------------|------------------|----------|
| 1000 | Poza 7 Svetloba 16 Izraz 5 | 640 x 480 | 52.000 |

Tabela 7: Statistika korejske podatkovne baze obrazov (KFDB).

Baza vsebuje veliko število slik korejskih subjektov, zajetih pod skrbno kontroliranimi pogoji. Subjekti so bili posneti z sedmimi kamerami, osvetljeni z osmimi reflektorji pred modrim ozadjem. Kamere so bile postavljene pod različnimi koti glede na sprednji položaj obraza (po loku navidezne krožnice) v razmiku 15° (do 45° desno od subjekta in do 45° levo od subjekta).

Podatkovna baza obrazov CVL

| Št. subjektov | Variacije | Ločljivost slike | Št. slik |
|---------------|-----------|------------------|----------|
| 114 | Poza 7 | 640 x 480 | 798 |

Tabela 8: Statistika podatkovne baze obrazov CVL.

CVL bazo obrazov je leta 1998 ustvaril dr. Peter Peer v laboratoriju za računalniški vid na Fakulteti za računalništvo in informatiko v Ljubljani. Baza zajema slike sto štirinajstih subjektov (7 slik za vsako osebo – glej sliko), večinoma moških (okoli 90%). Slike so bile zajete z digitalno kamero Sony Digital Mavica pod pogoji konstantne osvetlitve: brez bliskavice in s projekcijskim platnom v ozadju. Baza je namenjena nekomercialni uporabi in raziskovanju na področju računalniškega vida.

Dostopno na <http://www.lrv.fri.uni-lj.si/facedb.html>.

Povzetek javno dostopnih podatkovnih baz obrazov je narejen po knjigi Handbook of Face Recognition, avtorja Ralpa Grossa, v poglavju Face Databases (več v poglavju Literatura).



Slika 2: Variacije poz v bazi CVL.

1.3 Dela s podobno tematiko

V tem poglavju je predstavljen sorodni sistem za označevanje slik. Predstavljene so njegove prednosti in slabosti ter glavne razlike glede na rešitev, katero opisuje delo. Predstavljeni so tudi glavni razlogi, zakaj ta rešitev ni uporabljena za implementacijo predstavljenega problema.

1.3.1 Orodje LabelME

Pri iskanju rešitev za zastavljeni problem sem naletel na orodje LabelME (dostopno na <http://people.csail.mit.edu/brussell/research/LabelMe/intro.html>), ki na prvi pogled deluje kot idelna rešitev za zastavljeni problem. Orodje je odprto in ponuja uporabnikom zmožnost označevanja objektov na slikah. Razvito je bilo v

laboratoriju za umetno inteligenco na tehnološkem inštitutu v Massachusettsu (MIT). Avtorji dela so Bryan Russel, Antonio Torralba in William T. Freeman.

Kot je že omenjeno, je orodje namenjeno označevanju slik za uporabo označenih slik pri testiranju in izboljševanju algoritmov za detekcijo. Med orodjem LabelME in rešitvijo, opisano v tem delu, lahko potegnem veliko vspešnic. V grobem: orodje LabelME vsebuje naslednje funkcionalnosti (slika 3 prikazuje glavne module orodja):

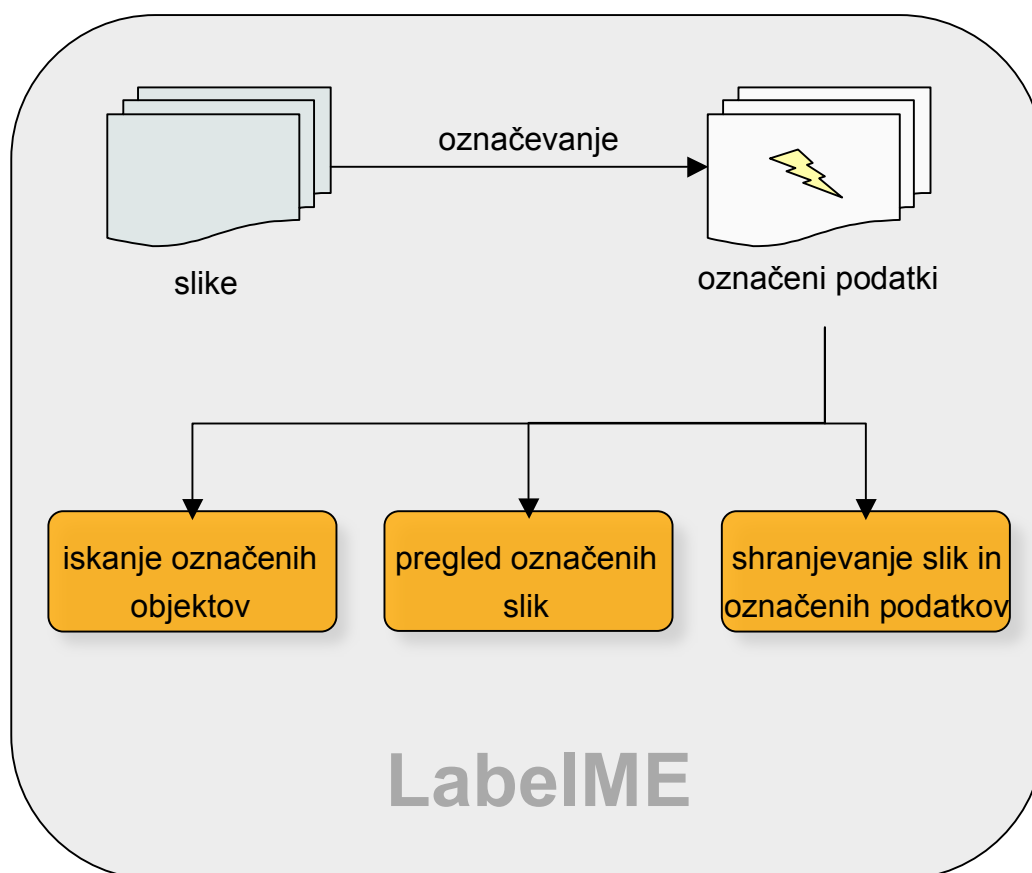
- modul za označevanje slik,
- pregled označenih slik po kategorijah,
- prenos baze slik in označb ter
- iskanje označenih objektov v bazi.

Predstavljeno orodje v tej nalogi obsega (poleg ostalih) iste funkcionalnosti, vendar pa je zaradi specifičnega cilja (anotiranje obrazov in analiza rezultatov algoritmov za iskanje in prepoznavo obrazov), zahtevalo razširitev nekaterih zgoraj naštetih funkcionalnosti. Cilji moje naloge so označevanje človeških obrazov, njihovih značilk, ter izbiranje različnih lastnosti obrazov in značilk. V ta namen bi bilo potrebno orodje LabelME razširiti z dodatnimi funkcionalnostmi pri samem označevanju slik. Druga večja pomankljivost orodja LabelME je nezmožnost integriranja in prikaza rezultatov algoritmov. Slednja funkcionalnost je ključnega pomena pri sami analizi rezultatov algoritmov.

Pri izdelavi moje naloge sem izbral iste tehnike in tehnologije za dosego cilja, kot so uporabljene pri izdelavi orodja LabelME. Uporabljene tehnologije sicer niso nove, so pa na novo uporabljene v praksi za implementacijo takšne funkcionalnosti. Označevanje samih slik poteka v realnem času prek spletnega vmesnika. Aplikacija, ki služi kot orodje za označevanje slik, je v obeh primerih spletni odjemalec. Da se takšna funkcionalnost implementira, je potrebno zaobiti veliko omejitev, ki jih postavljajo spletni odjemalci. Napisati je potrebno posebne knjižnice, s pomočjo katerih lahko z obstoječimi objekti simuliramo risanje oz. označevanje v spletnem odjemalcu. Pri testiranju in ocenjevanju orodja LabelME sem naletel na različne težave pri uporabi različnih spletnih odjemalcev. Ta ugotovitev me je še posebno motivirala za razvoj univerzalne rešitve, neodvisne od vrste spletnega odjemalca, kar se je kasneje izkazalo za resnično trd oreh.

Ne glede na specifične zahteve moje naloge, je bilo orodje LabelME edina dosegljiva in prava iztočnica za dosego cilja. Na tej točki bi lahko celo rekel, da je

ponujena rešitev neke vrste nadgradnja orodja LabelME. Ponuja iste funkcionalnosti, vendar le te razširja za namen reševanja zelo ozkega in specifičnega problema.



Slika 3: Shema orodja LabelME.

1.3.2 Ostala orodja

Obstajajo tudi druga orodja za označevanje slik, vendar nobeno od njih ne ponuja fleksibilnosti in funkcionalnosti, namenjenih specifično za označevanje obrazov, njegovih značilnk in njihovih lastnosti. Pregledal sem kar nekaj odprtokodnih orodij, primernih za anotacijo slik, vendar bi za njihovo integracijo v svoj sistem porabil precej več časa, kot sem ga porabil za izdelavo svojega orodja. V večini so to orodja, napisana v programskem jeziku Java, kar spet prinaša veliko omejitev pri delu s spletnim odjemalcem.

1.3.3 Ugotovitev

Pri pregledu podobnih del se odpira zanimiva ugotovitev. Obstaja zelo malo ali celo nič celovitih in javno dostopnih rešitev na področju označevanja človeških obrazov in sploh analize rezultatov algoritmov. Obstajajo različna orodja, ki

deloma rešujejo posamezne probleme, vendar nobeno od njih ne ponuja dobre rešitve izključno za področje izboljšanja algoritmov za razpoznavo obrazov.

2 □ Orodje za označevanje in analizo človeških obrazov na slikah

2.1 Cilji in zahteve

Prvi korak pri izdelavi orodja za označevanje in analizo človeških obrazov na slikah (v nadaljevanju FAAT – *Face Annotation and Analysis Tool*) je bila priprava seznama ključnih zahtev in ciljev. Ključne zahteve, od katerih mora implementacija orodja odstopati čim manj, so:

- delovanje prek spleta,
- uporaba tankega klienta (spletni odjemalec),
- neodvisnost od platforme (operacijskega sistema) ter
- uporaba odprtokodnih rešitev.

Po pregledu ključnih zahtev sistema je sledil zapis seznama ključnih ciljev. Gledano na splošno so okvir orodja sledeči cilji:

- zmožnost ročne označitve slik,
- zmožnost shranjevanja označb v primernem formatu,
- prenos podatkovne baze že označenih slik,
- integracija rezultatov algoritmov,
- prikaz rezultatov algoritmov,
- analiza rezultatov,
- administracija sistema.

Na podlagi razčlenitve zgornjih zahtev in ciljev sem se lotil pregleda in ocene obstoječih tehnologij, orodij in tehnik. Na podlagi ocen in po pregledu različnih pristopov sem izbral tehnologije, ki so po moji oceni optimalne za izdelavo predlagane rešitve. Ključne tehnologije, na katerih sloni implementacija orodja, so:

- strežnik na osnovi Linux operacijskega sistema,
- spletni strežnik Apache,
- podatkovna baza MySQL,
- skriptni jezik PHP in JavaScript,
- tehnika razvijanja spletnih aplikacij AJAX,
- označevalni jezik XML in

- modelirni jezik UML.

UML (*Unified Modeling Language*) jezik je bil uporabljen za razčlenitev osnovnih uporabniških funkcionalnosti orodja na več pod-funkcionalnosti na osnovi analize uporabniškega primera (use case). UML je računalniški standard, skupek elementov, ki standardizira modeliranje računalniških sistemov, procesov, aktivnosti, komunikacije med moduli sistema, itd. Analiza uporabniškega primera je tehnika pri razvoju programske opreme (programski inženiring) za zajem zahtev novega sistema ali morebitnih sprememb obstoječega sistema. Vsak uporabniški primer zajema enega ali več scenarijev, ki nam dajejo osnovno informacijo o tem kako naj bi bil sistem v interakciji s končnim uporabnikom ali drugim sistemom. Uporabniški primer se ponavlja izmika uporabi in opisovanju tehniškega vidika sistema.

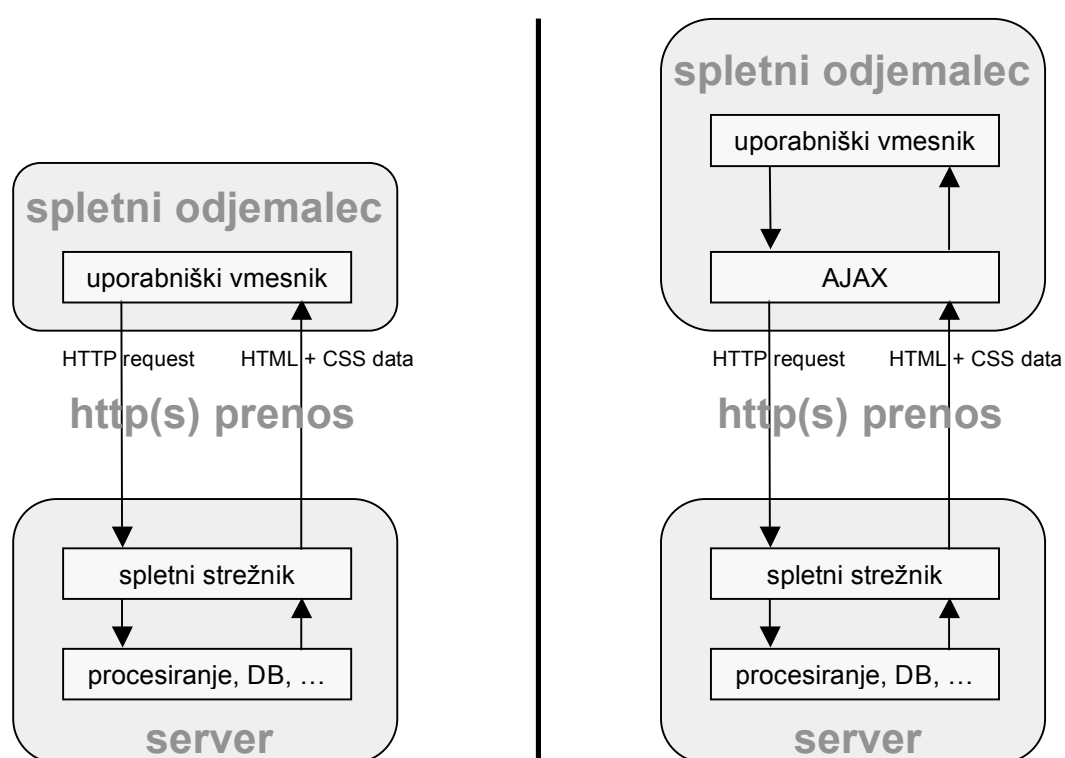
Na tem mestu bi omenil še uporabljeno tehniko AJAX. AJAX (angl. *Asynchronous JavaScript And XML*) ni tehnologija, je tehnika razvoja interaktivnih spletnih strani z uporabo več tehnologij:

- XHTML (ali HTML) in CSS jezik za predstavitev informacij,
- DOM objekt (*Document Object Model*) za dinamičen prikaz, dostopen s strani spletnega odjemalca (skriptni jezik JavaScript ali JScript),
- XMLHttpRequest objekt, s katerim dejansko izmenjujemo podatke med spletnim odjemalcem in spletnim strežnikom,
- XML jezik kot format prenosa podatkov med spletnim odjemalcem in spletnim strežnikom.

Osnovna ideja tehnike AJAX je izmenjava manjših paketov podatkov med spletnim odjemalcem in spletnim strežnikom. Klasični model spletne aplikacije deluje na sledeč način: Večina uporabniških akcij na spletni aplikaciji sproži HTTP zahtevo na spletni strežnik. Strežnik nato sprocesa podatke in spletnemu odjemalcu vrne novo HTML stran. Pristop s tehniko AJAX je malo drugačen (slika 4). Izmenjava podatkov je spet HTTP poizvedba spletnega odjemalca na spletni strežnik, vendar se ta poizvedba zgodi brez potrebe po osveževanju oz. ponovnem nalaganju celotne strani v primeru, ko uporabnik na strani sproži kakšno akcijo. V takšnem primeru spletni strežnik odjemalcu vrne samo del nove HTML strani (oz. nove informacije za prikaz), katerega potem z uporabo skriptnega jezika in DOM objekta dinamično prikažemo na strani spletnega odjemalca.

Pristop po klasičnem modelu spletnih aplikacij ima velik tehnični smisel, ne pripomore pa veliko k dobri uporabniški izkušnji. Kaj dela uporabnik medtem, ko server procesira podatke? Čaka. In na vsakem novem koraku svojega opravka spet čaka.

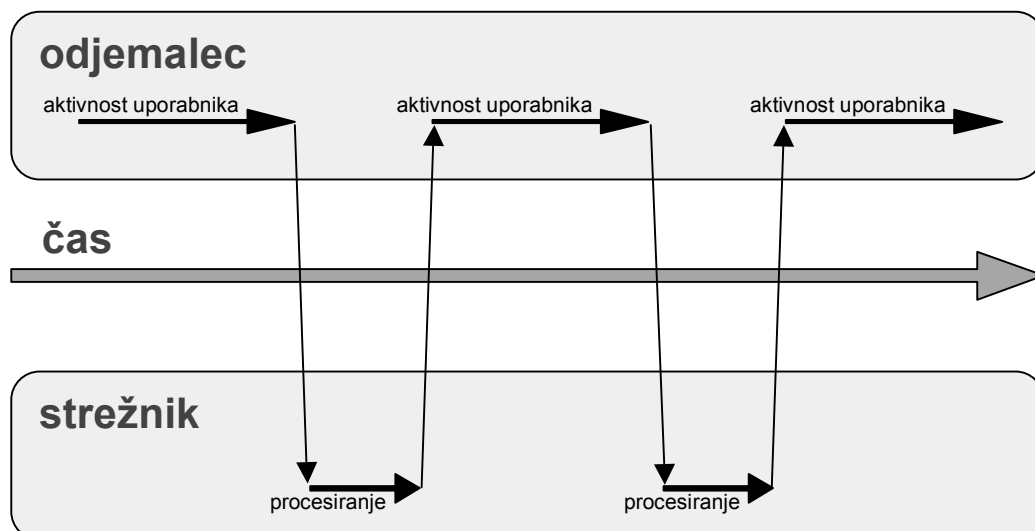
Z uporabniškega vidika je očitno, da ne želimo ustvariti spletne aplikacije, pri kateri bi uporabnik predolgo čakal na informacije in s tem izgubljal dragoceni čas. Zakaj bi bilo potrebno ponovno nalaganje uporabniškega vmesnika vsakič, ko aplikacija potrebuje nekaj iz strežnika, s čimer postaviti uporabnika na čakanje? Pravzaprav, zakaj bi moral uporabnik za takšno interakcijo sploh vedeti? In prav z uporabo AJAX tehnike se lahko vsemu temu v veliki meri izognemo.



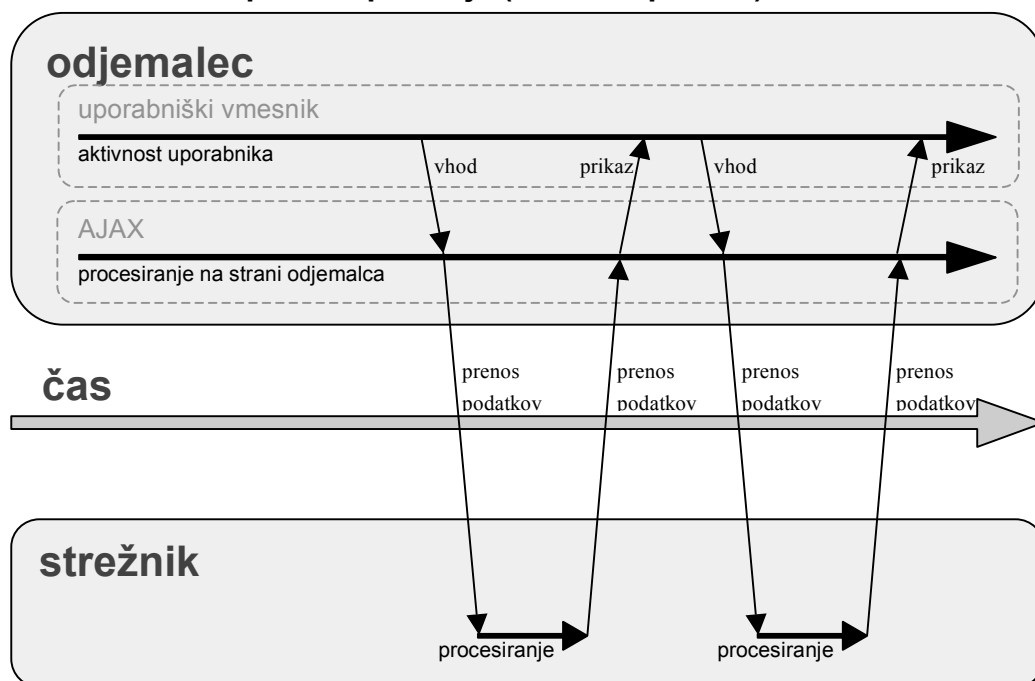
Slika 4: Klasični model spletne strani (levo) in AJAX model spletne strani (desno).

AJAX tehnika je torej predvsem uporabljena za povečanje interaktivnosti spletnih strani, hitrosti in izboljšanje uporabnosti. Pri nalogi je tehnika uporabljena za branje predlog XML datotek, ki so nadalje uporabljene za shranjevanje podatkov anotacije na strežnik. Povzetek razvoja spletnih aplikacij s tehniko AJAX sem povzel iz članka J. J. Gareth-a, objavljenega na spletnem portalu Adaptive Path Publications (podrobnosti v poglavju Literatura).

Klasični model spletne aplikacije (sinhroni prenos)



AJAX model spletne aplikacije (sinhroni prenos)



Slika 5: Razlika med različnima modeloma spletnih strani.

2.2 Načrtovanje orodja

Načrtovanje orodja se je začelo z zbiranjem ključnih zahtev in funkcionalnosti, za katere smo želeli, da jih zajema. Pred samim začetkom implementacije je bila potrebna analiza dostopnih tehnik in tehnologij (glej poglavje 3.1 za uporabljene tehnologije in tehnike), s katerimi bi si zagotovili najboljše razmerje med enostavno implementacijo in prijaznostjo sistema do uporabnika. Ker podobne

rešitve skoraj ne obstajajo, je bila pomembnost koraka analize tehnologij in tehnik, ki sem jih kasneje uporabil, toliko večja. Sprememba tehnologije (na primer programskega jezika) v fazi same implementacije za sabo potegne veliko problemov, predvsem pa je ponavadi vzrok za velike časovne zamude pri končanju nekega projekta.

2.2.1 Analiza uporabniškega modela

Modelirni jezik UML (omenjen v poglavju 3.1) je standard, uporabljen za analizo uporabniškega primera orodja FAAT. Spisek ključnih zahtev in funkcionalnost služita kot osnova za nastanek takšnega primera.

Ključne zahteve in funkcionalnosti so razbite na štiri različne uporabnike orodja (akterje) in sestavljajo štiri različne scenarije:

- scenarij 1 - uporabnik označevanja (annotation user),
- scenarij 2 - uporabnik podatkovne baze (database user),
- scenarij 3 - uporabnik analize (analysis user) in
- scenarij 4 - administrator sistema.

Ti različni tipi uporabnikov nadalje interaktivno sodelujejo z ostalimi moduli sistema. To interakcijo med uporabniki in sistemom opisuje analiza uporabniškega primera (prikazana na slikah 6 – 9).

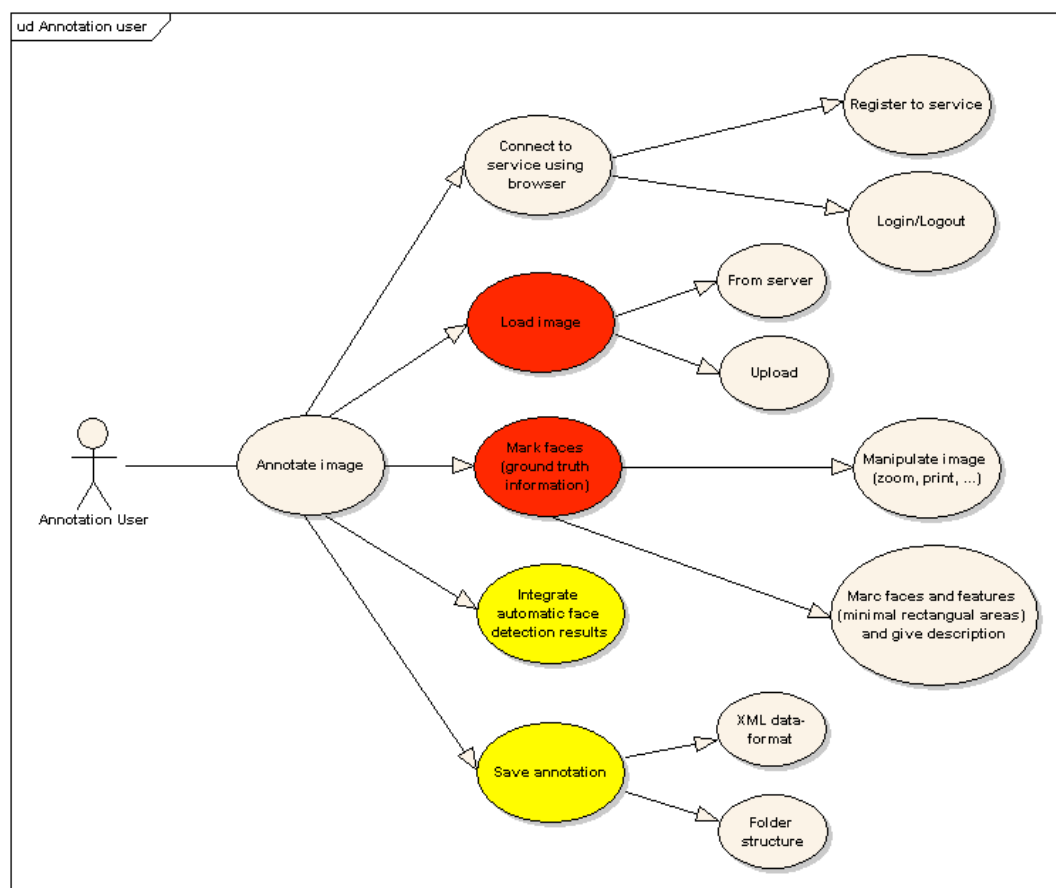
2.2.1.1 Uporabnik označevanja

Scenarij 1 (slika 6) opisuje interakcijo med sistemom in uporabnikom označevanja. Uporabnik označevanja je uporabnik z zmožnostjo označevanja slik, torej označevanja obrazov, njegovih značilk in njihovih lastnosti ter uporabo ostalih funkcij, potrebnih za izvršitev operacije označevanja. Uporabniški model za scenarij 1 zajema naslednje funkcionalnosti:

- prijava v sistem,
- nalaganje slike,
- označevanje obrazov, njegovih značilk in njihovih lastnosti,
- integracija rezultatov avtomatske detekcije obrazov in
- shranjevanje anotacije.

Prijava v sistem je funkcionalnost avtentikacije in prijave uporabnikov v aplikacijo za označevanje slik. Poleg prijave in odjave v aplikacijo se uporabniški primer členi še na:

- prijavo novega uporabnika in
- ponovno pošiljanje izgubljenega gesla na elektronski naslov uporabnika.



Slika 6: Uporabniški model za scenarij 1 – uporabnik anotacije.

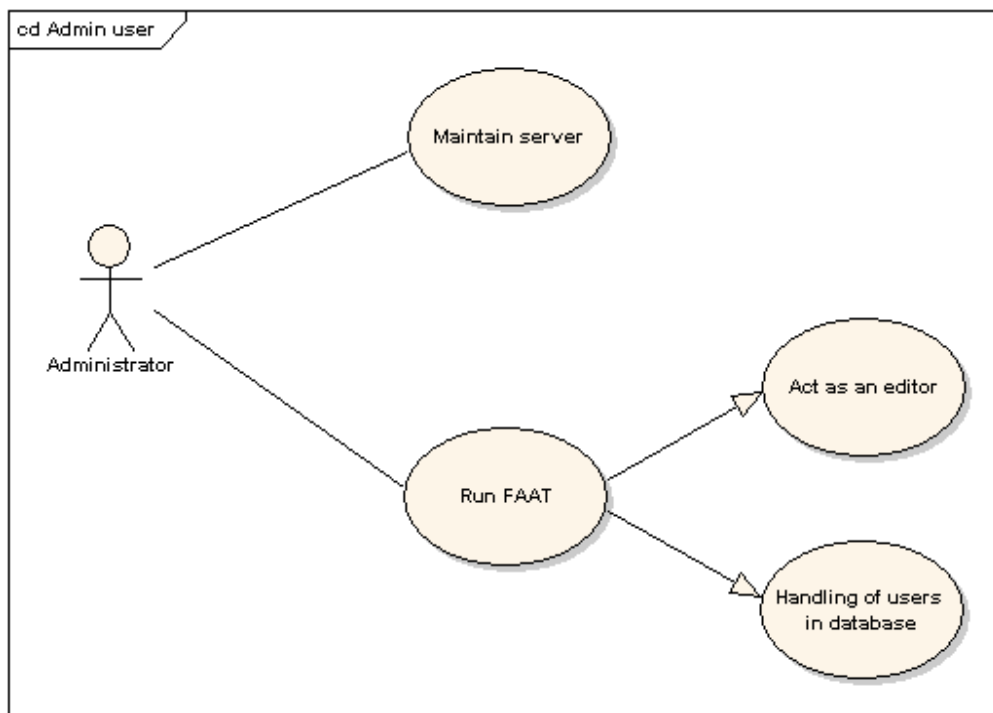
2.2.1.2 Administrator sistema

Drugi scenarij analize uporabniškega modela (slika 7) opisuje interakcijo med uporabnikom administratorjem in sistemom. Administrator je uporabnik z največjo stopnjo pravic, ki skrbi za vzdrževanje sistema. Model scenarija smo razčlenili na dve ključni funkcionalnosti:

- vzdrževanje strežnika in
- upravljanje s sistemom.

Vdrževanje sistema zajema upravljanje z datotekami na strežniku in ostala vzdrževalna dela (nastavitve sistema, zagon sistema, vzdrževalna dela na

podatkovni bazi, itd.). Druga ključna funkcionalnost tega scenarija je upravljanje s sistemom. Tukaj uporabnik skrbi za upravljanje z uporabniki v podatkovni bazi in deluje kot neke vrste urednik. Pod to nalogo uporabnik (administrator) pregleduje in po potrebi popravlja novo označene slike. Ima možnost zavračanja nepravilno označenih slik oz. potrjevanja njihove veljavnosti.

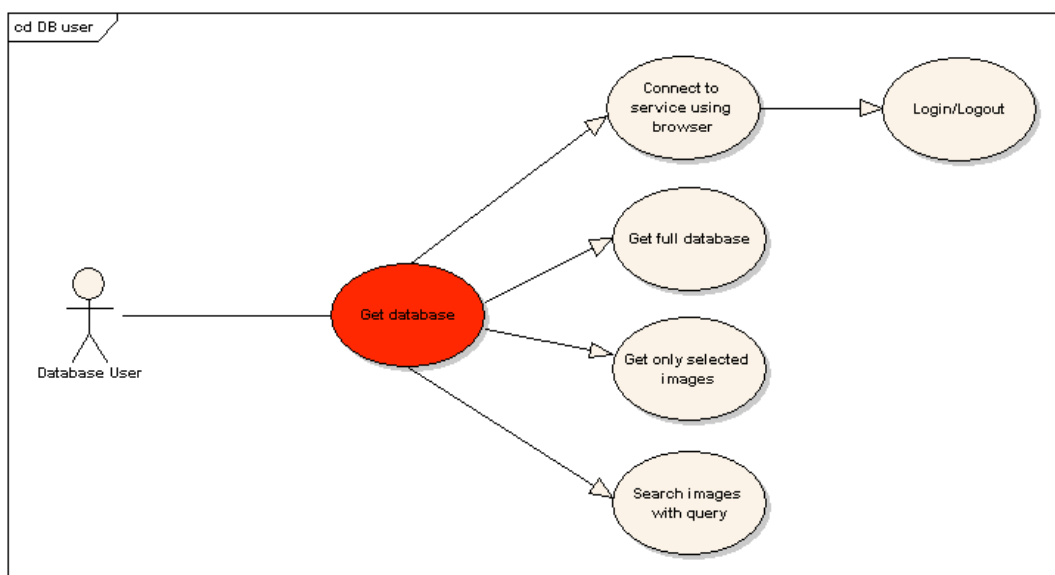


Slika 7: Uporabniški model za scenarij 2 – administrator.

2.2.1.3 Uporabnik podatkovne baze

Slika 8 prikazuje analizo uporabniškega modela za tretji možni scenarij – uporabnik podatkovne baze. Scenarij opisuje možnost prenosa podatkovne baze (slik in pripadajočih podatkov označb) na računalnik uporabnika. Model prenosa podatkovne baze smo razdelili na:

- prijavo uporabnika v sistem,
- pridobivanja celotne podatkovne baze,
- pridobivanja podatkovne baze z izbiranjem slik in
- pridobivanja podatkovne baze z iskalnimi kriteriji.

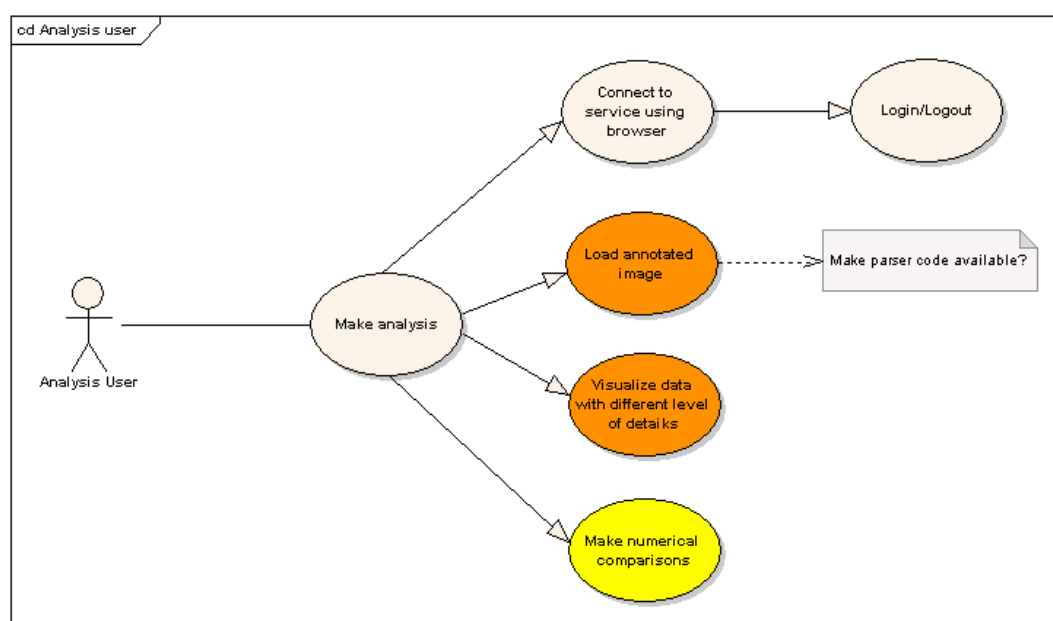


Slika 8: Uporabniški model za scenarij 3 – uporabnik podatkovne baze.

2.2.1.4 Uporabnik analize

Scenarij opisuje uporabnika analize oz. potrebne funkcionalnosti za uspešno analizo rezultatov algoritmov. Uporabniški primer (slika 9) sem razčlenil na naslednje funkcionalnosti:

- prijava v sistem,
- nalaganje označene slike,
- prikaz podatkov z različno stopnjo podrobnosti,
- možnost numeričnih primerjav.



Slika 9: Uporabniški model za scenarij 4 – uporabnik analize.

2.2.2 Sklep

Slike 6 – 9 prikazujejo analizo primerov uporabe, štiri možne scenarije in interakcijo med moduli ter uporabniki. Različne barve posameznih objektov podajajo prioriteto implementacije. Tako rdeča barva pomeni stopnjo največje prioritete, sledijo ji oranžna, rumena in svetlo rjava, kot barva z najmanjšo prioriteto. To seveda ne pomeni, da funkcionalnosti z nižjo prioriteto niso pomembne. Pomeni le, da se funkcionalnosti z največjo prioriteto implementirajo prej, saj ima smisel implementacijo graditi na najbolj ključnih in obsežnih funkcionalnostih.

Posamezne funkcionalnosti se naprej členijo na pod-funkcionalnosti in te nadalje na nove pod-funkcionalnosti, itd. Analiza našega modela vsebuje maksimalno tri stopnje razčlenjevanja funkcionalnosti na pod-funkcionalnosti, saj nima smisla komplicirati in členiti model v prevelike podrobnosti. S preveč podrobnim razčlenjevanjem izgubimo na preglednosti in berljivosti uporabniških primerov, s čimer lahko izgubimo rdečo nit takšne analize.

Poleg analize uporabniških primerov bi bilo pred samim pričetkom implementacije takšnega sistema v praksi dobro premisliti še o ostalih analizah. Glede na obsežnost sistema bi bilo dobro razmisliti o pripravi dokumenta z načrtom programske opreme (angl. *design document*). Dokument praviloma pripravi načrtovalec programske opreme (ali sam razvijalec), da začrta smernice arhitekture programske opreme. Podajal naj bi kompleten opis načrtovanja programske opreme, ki pa mora ohranjati visoko-nivojski pregled nad programsko opremo. Ponavadi vanj vključimo diagrame arhitekture in načrte implementacije posameznih modulov. Dokument je praktično nujen za koordinacijo ekipe razvijalcev, še posebej, če se implementacija deli med več razvijalcev. V praksi poznamo dve vrsti tega dokumenta: visoko nivojski (angl. *high level design document*) in nižje nivojski (angl. *low level design document*). Opis dokumenta z načrtom programske opreme sem povzel iz spletne enciklopedije Wikipedia (podrobnosti v poglavju Literatura).

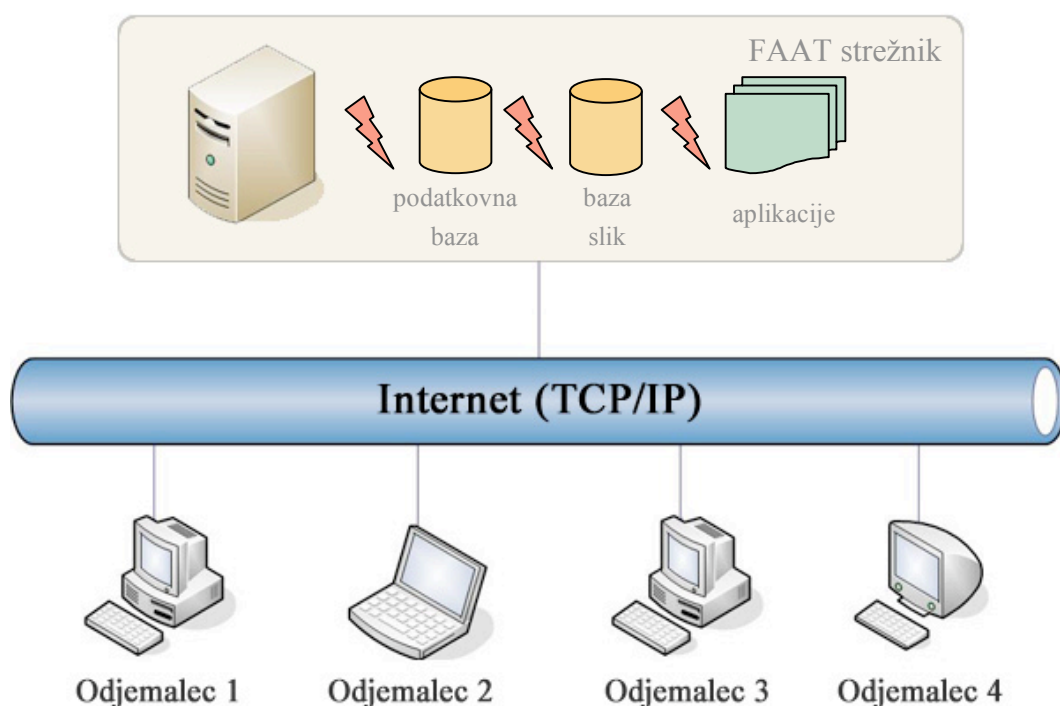
2.3 Arhitektura orodja FAAT

Poglavje opisuje strukturo orodja FAAT in njegove ključne komponente. Že iz analize uporabniških primerov v prejšnem poglavju je razvidno, da orodje sestoji iz več manjših aplikacij in ostalih samostojnih komponent. Arhitektura orodja je modularna in dopušča ponovno uporabo posameznih modulov. Takšna zasnova zmanjšuje problem podvajanja kode in s tem podvajanja napak.

Orodje FAAT lahko torej povzamemo kot skupek naslednjih komponent (slika 10):

- avtentikacija uporabnikov,
- aplikacija za označevanje slik,
- aplikacija za administriranje sistema,
- aplikacija za prenos baze slik,
- aplikacija za analizo rezultatov,
- podatkovna baza,
- repozitorij (baza) slik in podatkov označb

Arhitektura strežnik – odjemalec omogoča fleksibilnost in neodvisnost odjemalcev od same logike in podatkovne baze. Število odjemalcev je omejeno na zmogljivost strežnika in ostale opreme (podatkovna baza, povezava, itd...), vendar zaradi narave problematike in verjetno majhnega števila sočasnih uporabnikov ne predstavlja nekih performančnih problemov, zaradi katerih bi bilo potrebno sistem razširiti z ustrežno strojno in programsko opremo.



Slika 10: Arhitektura orodja FAAT.

Trenutna namestitvev sistema zajema namestitvev spletnega strežnika, podatkovne baze, baze slik in aplikacij na isti strežnik. Mogoča je ločitev komponent po več strežnikih. V tem primeru posamezni odjemalci komunicirajo direktno z

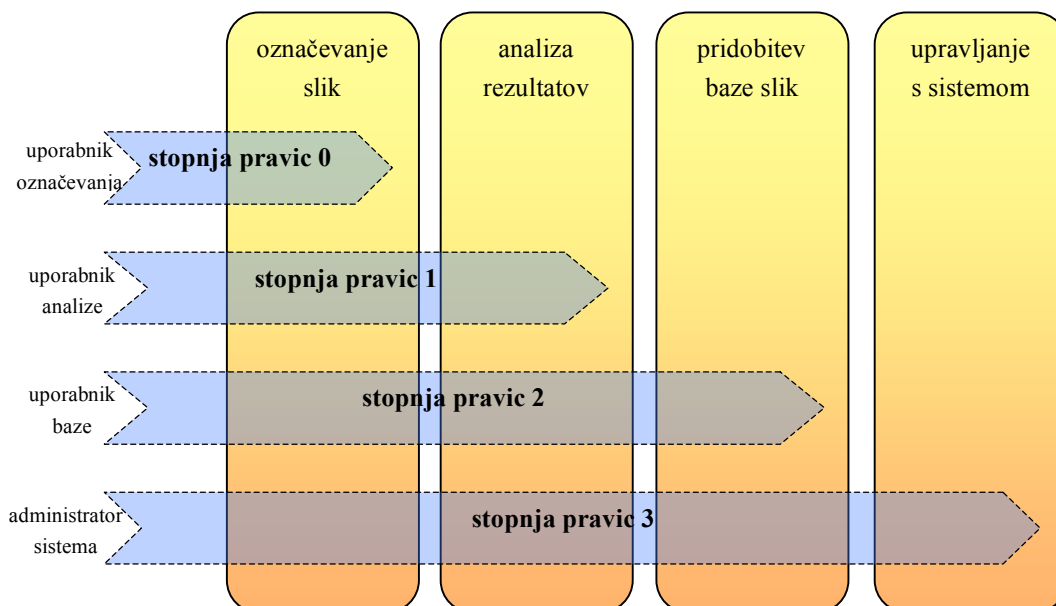
aplikacijskim strežnikom, ta pa naprej z ostalimi strežniki. Za takšno namestitev je potrebno ustrezno nastaviti konfiguracijo na aplikacijskem strežniku (podrobnosti v poglavju 3.6).

2.3.1 Avtentikacija uporabnikov

Za delo z aplikacijami orodja FAAT mora uporabnik pridobiti pravice za njihovo uporabo. Za pridobitev in upravljanje s pravicami skrbi posebna aplikacija ali modul za avtentikacijo uporabnikov. V poglavju 3.2.1. (analiza uporabniških primerov) smo spoznali različne scenarije oz. različne tipe uporabnikov. Če še enkrat naredim povzetek, lahko v sistemu delujejo naslednji tipi uporabnikov:

- administrator sistema,
- uporabnik označevanja,
- uporabnik podatkovne baze slik,
- uporabnik analize.

Vsak od naštetih tipov uporabnikov ima svojo stopnjo pravic. S stopnjo pravic uporabnik pridobi dovoljenje za uporabo določenih funkcij orodja. Dodeljevanje pravic poteka avtomatsko, samo preverjanje pa izvaja modul za avtentikacijo uporabnikov. Slika 11 prikazuje različne stopnje pravic in povezavo med stopnjo pravic, vrst uporabnikov ter aplikacijami.



Slika 11: Shema uporabniških pravic sistema.

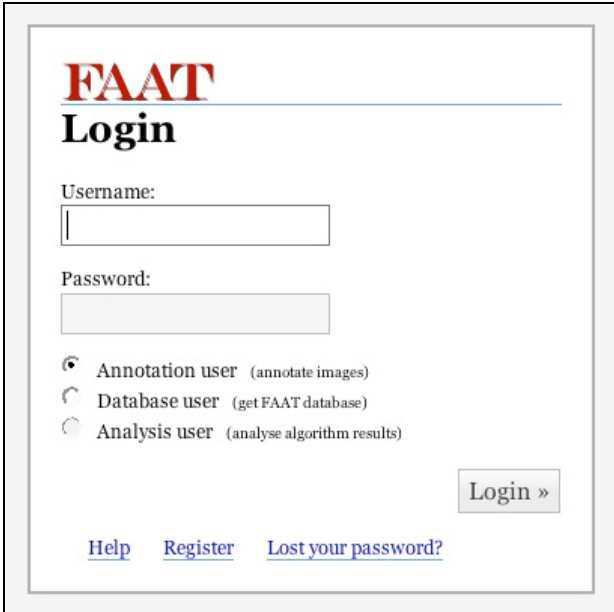
Kot vidimo iz sheme, imajo različni tipi uporabnikov različne pravice. Za pridobitev določenih pravic je potrebno razložiti sistem pridobivanja pravic. Ob prijavi novega uporabnika le ta pridobi pravico uporabnika označevanja (stopnja pravic 0). S tem je uporabniku omogočena osnovna funkcionalnost orodja – označevanje slik.

S številom označenih slik uporabniku rastejo pravice. Uporabnik označevanja pridobi pravice uporabnika analize (stopnja pravic 1) po petih uspešno označenih slikah. Uspešno označena slika ni vsaka, ki jo uporabnik označi, ampak le tista katero je urednik (administrator sistema) po pregledu potrdil. Isto velja za pridobitev pravice uporabnika baze (stopnja pravic 2), le da v tem primeru šteje deset uspešno označenih slik. V primeru, ko uporabnik doseže stopnjo pravic 2 lahko upravlja vse funkcije orodja FAAT, z izjemo administracije. Administracijo lahko opravlja samo administrator sistema. Privzeto je s postavitvijo sistema na strežnik avtomatsko kreiran administrator sistema, vendar je naknadno možno prijaviti tudi več administratorjev sistema. Te lahko prijavi privzeto kreirani administrator in to samo z ročnim vnosom v bazo ali s kreiranjem skripte.

2.3.1.1 Funkcionalnost in uporabnost

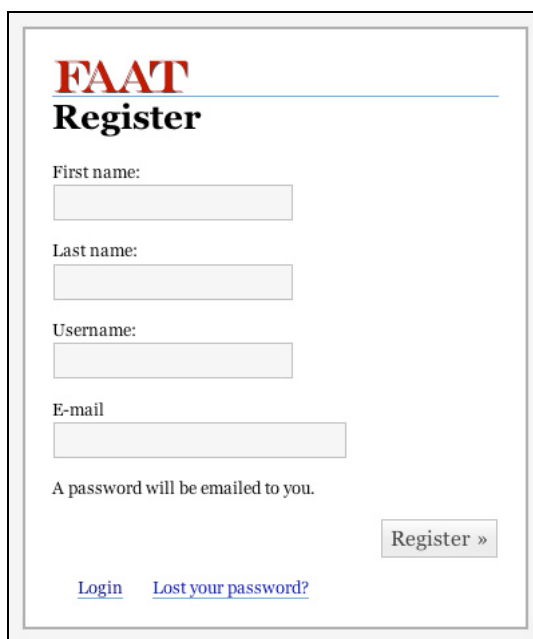
Slika 12 prikazuje uporabniški vmesnik avtentikacije uporabnikov. Poleg same avtentikacije uporabnikov modul izvaja še:

- funkcijo registracije novih uporabnikov in
- funkcijo pošiljanja izgubljenega gesla.



Slika 12: Avtentikacija uporabnikov.

Ob prijavi uporabnik izbere tip uporabnika, aplikacija preveri ustreznost uporabniških pravic in ga nato prijavi v sistem. Prijava je v primeru neustreznih pravic neuspešna. Avtentikacija uporabnika se vrši glede na izbrano uporabniško ime in geslo. Uporabnik ob registraciji izbere željeno uporabniško ime (slika 13). Pri tem med drugim navede tudi svoj veljavni elektronski naslov. Sistem nato avtomatsko ustvari geslo in pošlje vse potrebne podatke za prijavo na navedeni elektronski naslov.

The image shows a web form titled "FAAT Register". The form contains several input fields: "First name:", "Last name:", "Username:", and "E-mail". Below the "E-mail" field, there is a text line that says "A password will be emailed to you." At the bottom right of the form is a button labeled "Register »". At the bottom left, there are two links: "Login" and "Lost your password?". The form is enclosed in a simple rectangular border.

Slika 13: Registracija uporabnikov.

2.3.2 Aplikacija za označevanje slik

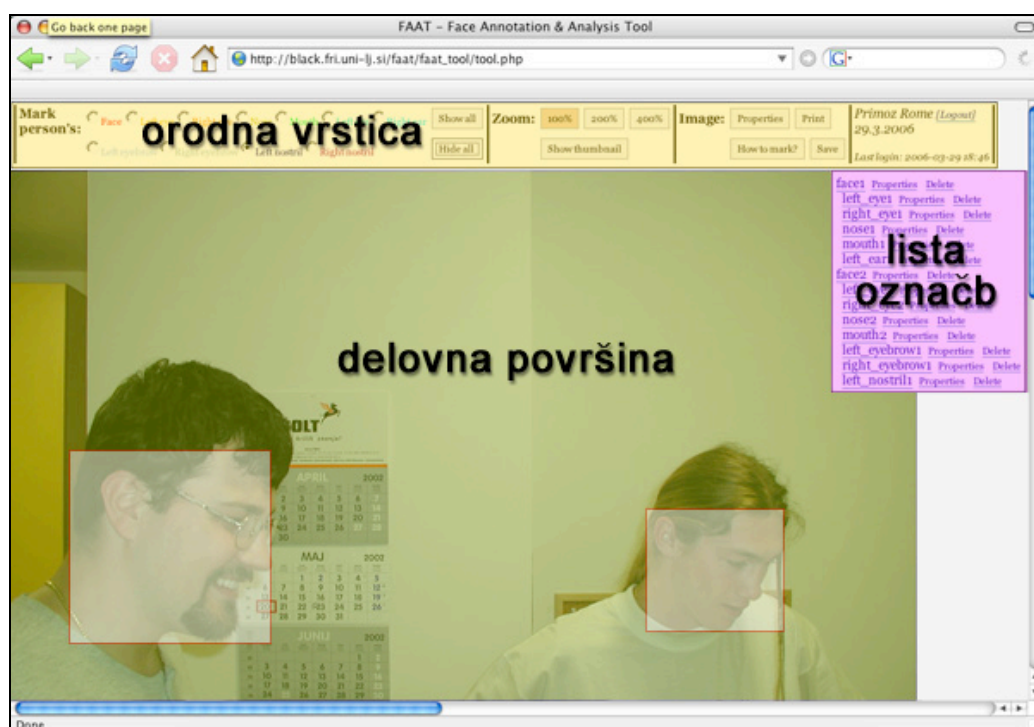
Aplikacija za označevanje slik je najobsežnejša aplikacija orodja in je namenjena ročnemu označevanju obrazov, njegovih značilk in njihovih lastnosti. Je spletna aplikacija (angl. *web application*) in podpira večino spletnih odjemalcev nove generacije. Zaradi nekaterih specifičnih gradnikov ima aplikacija še vedno težave s kompatibilnostjo, predvsem s starejšimi verzijami odjemalcev. Treutna lista delujočih in testiranih odjemalcev je sledeča:

- Microsoft Internet Explorer 5.0 ali novejši,
- Mozilla kompatibilni odjemalec (Firefox, Netscape, Camino, itd...) in
- Opera.

2.3.2.1 Funkcionalnost in uporabnost

V poglavju 3.2.1. smo že prikazali uporabniške primere aplikacije (slika 5). Uporabnik lahko označuje obstoječe slike iz repozitorija (baze) slik (lociran na strežniku) ali lastne slike naložene iz njegovega računalnika. Uporabniški vmesnik aplikacije (slika 14) se deli na tri področja:

- orodna vrstica,
- delovna površina,
- seznam označb.



Slika 14: Uporabniški vmesnik aplikacije za označevanje.

Označevanje slike poteka v naslednjem vrstnem redu:

1. izbira ustreznega tipa markerja (obraz ali izbrana značilka) v orodni vrstici,
2. označitev regije na sliki z izbranim markerjem,
3. izbira lastnosti za izbran obraz oz. značilko.

Tip markerja je lahko obraz ali značilka obraza. Značilke obraza se delijo na:

- levo in desno oko,
- levo in desno uho,
- nos,

- usta,
- levo in desno obrv ter
- levo in desno nosnico.

Vsakemu obrazu ali značilki uporabnik izbere lastnosti, ki so lahko obvezne ali opcijske. Sliki 15 in 16 prikazujeta seznam možnih lastnosti obraza in značilk.

od Face properties

Face properties

Gender

☐ + Female

☐ + Male

Face characteristics

☐ + Beard

☐ + Glasses

☐ + Makeup

☐ + Mustaches

Complexion

☐ + Black

☐ + Brown

☐ + Red

☐ + White

☐ + Yellow

Face orinetation

☐ + Frontal

☐ + Left profile 45° (left cheek)

☐ + Left profile 90° (left cheek)

☐ + Right profile 45° (right cheek)

☐ + Right profile 90° (right cheek)

Face expression

☐ + Anger

☐ + Disgust

☐ + Fear

☐ + Happy

☐ + Neutral

☐ + Sad

☐ + Surprise

Age

☐ + 0-2 years

☐ + 19-60 years

☐ + 3-18 years

☐ + Above 60 years

Hair color

☐ + Black

☐ + Blond

☐ + Brown

☐ + Gray

☐ + Non-conventional

☐ + Red

Hair length

☐ + Long

☐ + Medium

☐ + None

☐ + Short

Face skin

☐ + Clear

☐ + Frackled

☐ + Wrinkled

Comment

Properties with yellow color are mandatory

Properties with white color are optional

Slika 15: Shema lastnosti obraza.

cd Eye properties

Eye properties

Condition

+ Closed

+ Opened

Comment

Mouth properties

Condition

+ Closed

+ Opened

Teeth?

Comment

Nose, ear, eyebrow, nostril properties

Comment

Properties with yellow color are mandatory

Properties with white color are optional

Eye properties valid for left and right eye

Slika 16: Shema lastnosti obraznih značilk.

2.3.3 Aplikacija za prenos baze slik

2.3.3.1 Funkcionalnost in uporabnost

Aplikacija je namenjena pridobitvi baze slik iz strežnika (slika 17). Za uporabo aplikacije potrebuje uporabnik določeno stopnjo pravic (podrobnosti v poglavju 3.3.1.). Na voljo so tri metode za prenos baze:

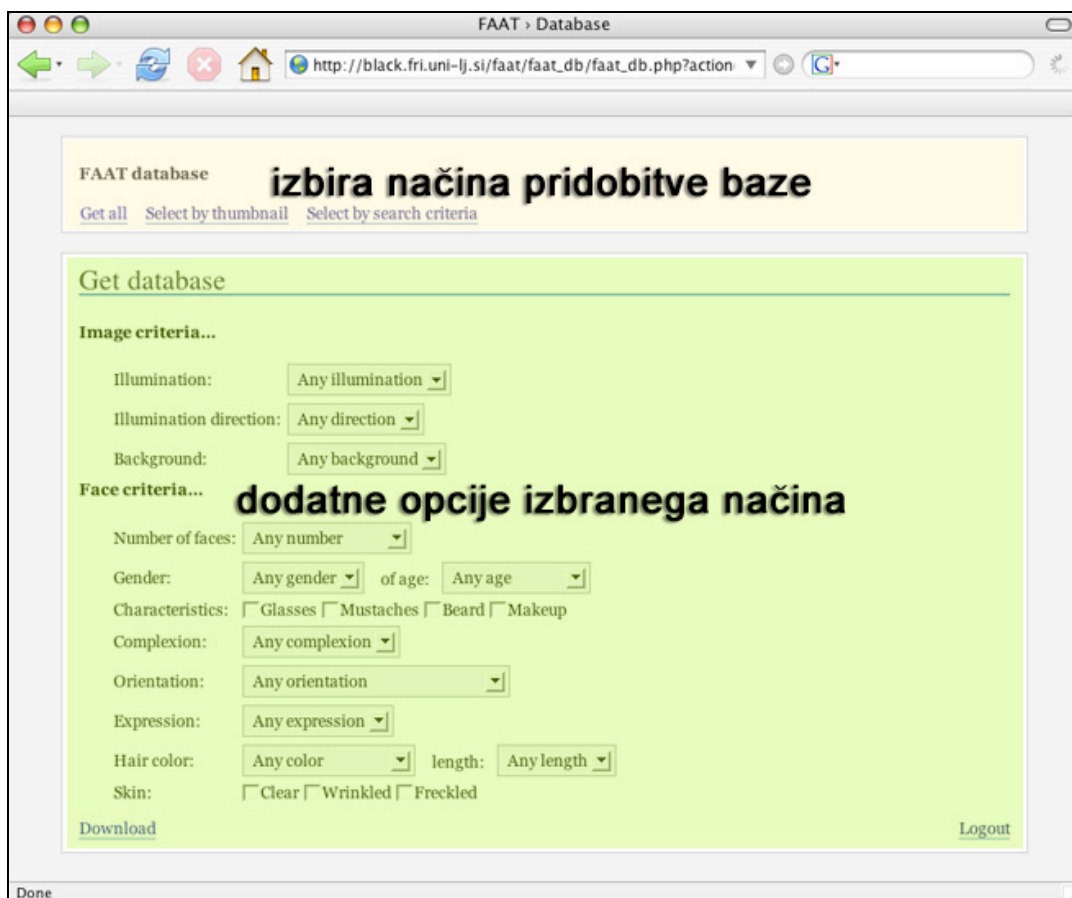
- prenos celotne baze,

- prenos samodejno izbranih slik in
- prenos slik na podlagi iskalnih kriterijev.

V prvem primeru uporabnik na računalnik prenese celotno bazo slik in njihove označbe. V drugem primeru aplikacija prikaže seznam obstoječih slik. Uporabnik iz seznama izbere slike, katere želi prenesti na svoj računalnik. V primeru tretje metode aplikacija od uporabnika zahteva vnos iskalnih kriterijev. Uporabnik prenese vse slike, ki ustrezajo iskalnim kriterijem.

Ne glede na izbrano opcijo se uporabniku baza prenese na računalnik v obliki ZIP datoteke. ZIP datoteka vsebuje slike in XML datoteke z označbami. Vsaki sliki pripada izbrana XML datoteka z označbami (poglavje 3.4 opisuje strukturo te datoteke). Primer: sliki *testna_slika.jpg* pripada istoimenska datoteka XML s podatki označb na tej sliki – *testna_slika.xml*.

Vsak uporabnik ob prvem prenosu podatkovne baze sprejme pogoje licence. V kolikor se uporabnik z njimi ne strinja, in to ustrezno označi, se prenos podatkovne baze prekine.



Slika 17: Aplikacija za prenos baze slik.

2.3.4 Aplikacija za analizo rezultatov

Možnost analize rezultatov algoritmov je ključna funkcionalnost sistema. Ostala podobna orodja takšne funkcije ne ponujajo. Načrtovalci algoritmov so prepuščeni samodejni analizi rezultatov. Prednost aplikacije je tudi v tem, da so vsi potrebni podatki in rezultati algoritmov vidni vsem uporabnikom sistema. Takšna zasnova omogoča hitro razširjanje informacij in znanja na področju razvoja algoritmov.

2.3.5 Funkcionalnost in uporabnost

V osnovi se definira odstotek zahtevanega prekrivanja med ročno vnešenimi podatki (angl. *ground truth data*) in rezultati algoritma. Oboji podatki so uporabniku vidni na zaslonu. Možen je prikaz podatkov po različnih plasteh (prikaz samo ročno označenih podatkov ali samo prikaz algoritmov).

Na izbrani sliki lahko ilustrirano prikažemo naslednje rezultate (povzeto po članku Color-Based Face Detection in the »15 Seconds of Fame« Art Installation avtorjev F. Solina, P. Peer, B. Batagelj, S. Juvan, J. Kovač. Podrobnosti v poglavju Literatura):

- pravilno detektiranih obrazov in značilk (angl. *True positives - TP*),
- nepravilno detektiranih obrazov in značilk (angl. *False positives - FP*) in
- zgrešenih obrazov in značilk (angl. *False negatives - FN*).

Iste podatke lahko dobimo na množici slik, vendar pa so le te prikazani numerično v tabeli.

2.4 FDRML

2.4.1 Uvod

FDRML je kratica za jezik za označevanje rezultatov detekcije obrazov (angl. *Face Detection Results Markup Language*), ki smo ga razvili. Jezik opisuje množico podatkov, dobljenih z ročnim označevanjem slik ter z rezultati algoritmov. Definirati je bilo potrebno lahko berljiv in pregleden format. Odločili smo se za zapis v XML formatu, ker je le ta:

- lahko berljiv tako človeku kot računalniškim sistemom,

- omogoča predstavitev najosnovnejših podatkovnih struktur (zapisi, drevesa, sezname, tabele),
- samo-opisni format, ki opisuje tako strukturo in imena polj kot njihove vrednosti,
- format je platformsko neodvisen, robusten in standardiziran.

XML je splošno označevalni jezik, namenjen opisovanju različnih struktur podatkov, in je standardiziran s strani W3C organizacije (World Wide Web Consortium). Primarni namen formata je prenos in izmenjava podatkov med različnimi sistemi, predvsem sistemi, ki so povezani preko interneta.

Pred samo implementacijo formata je bilo potrebno sestaviti model vseh željenih podatkov, definirati njihov tip in logično strukturo njihovega zapisa. V ta namen smo sestavili praktični model (slika 18), ki je nadalje služil kot osnova pri implementaciji XML sheme. XML shema je prav tako W3C standard, namenjen opisu XML jezika. Rezultat takšne sheme je XSD datoteka (angl. *XML Schema Definition*) z množico pravil, katerih se mora XML zapis držati, da ostane veljaven. Za izdelavo XML sheme smo uporabili orodje XML Spy (dostopno na <http://www.altova.com>).

Shema je javno objavljena na spletu in je dostopna na naslovu http://black.fri.uni-lj.si/faat/faat/schema/FDRML_schema.xsd. Uporabimo jo pri kreiranju novega XML dokumenta z jezikom FDMRL, in sicer tako, da to referenco na shemo dodamo v glavo dokumenta, tako kot prikazuje naslednji primer:

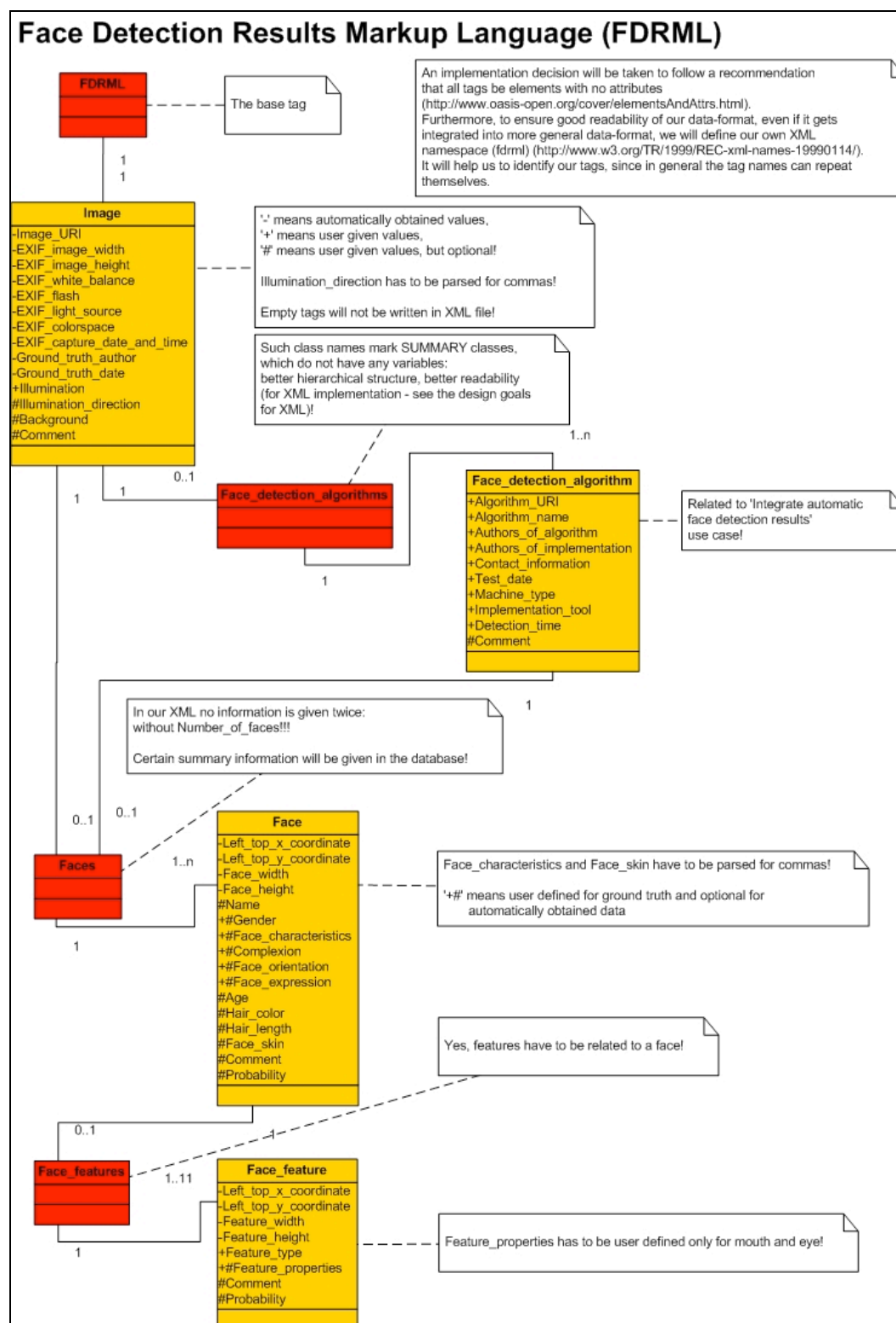
```
<FDMRL xsi:schemaLocation="fdrml http://black.fri.uni-lj.si/faat/faat/schema/FDRML_schema.xsd" Version="1.0">
```

Priloga B prikazuje izvorno kodo izdelane XML sheme.

2.4.2 Struktura in zapis podatkov

Struktura podatkov je organizirana drevesno. Deblo drevesa sestavljajo podatki obravnavane slike. Te podatke delimo na meta podatke slike in podatke definirane s strani uporabnika. Meta podatki slike so podatki, ki jih sistem prebere iz glave formata slike. Meta podatki so zapisani v standardnem formatu, imenovanem EXIF (angl. *Exchangable Image File format*). EXIF format je standardizirana specifikacija za format slik (JPEG, TIFF), zajetih z digitalnimi kamerami, in obsega podatke kot so datum in ura zajema slike, dimenzija slike, izvor svetlobe,

tip in model kamere, itd. Ostali podatki o obravnavani sliki so podatki, ki jih poda uporabnik prek uporabniškega vmesnika (osvetlitev slike, izvor osvetlitve, ozadje, itd.) oz. jih sistem zajame avtomatsko (URI slike, avtor označitve, datum in čas označitve).



Slika 18: Model strukture podatkov jezika FDRML.

Na stopnji podatkov obravnavane slike se struktura zapisa razveja na podatke o rezultatih algoritmov in na podatke o ročnih označbah slik. Iz slike 18 je razvidno, da lahko zapis ene slike zajame rezultate več algoritmov in več označb. Podatke rezultatov algoritma delimo na globalne podatke algoritma in na podatke najdenih in/ali prepoznanih obrazov in njihovih značilk, medtem ko podatke o ročnih označbah delimo samo na podatke o označenih obrazih in njegovih značilkah. Struktura in format podatkov o obrazih in njihovih značilkah je na obeh vejah ista, vendar pa ni nujno, da se le ti ujemajo. Podatki na veji o ročnih označbah so vedno vsi podatki podani s strani uporabnika v trenutku, ko je ta označil sliko z aplikacijo za označevanje obrazov. Podatki na veji o rezultatih algoritma so v sam zapis integrirani kasneje iz datoteke z rezultati algoritma (poglavje 3.5).

Priloga A prikazuje format izdelane XML datoteke.

2.5 Rezultati algoritmov

Rezultati algoritmov so podani v tekstovnih datotekah po vnaprej definiranem fiksnem formatu. Ena tekstovna datoteka opisuje rezultate enega algoritma na množici slik, na kateri je algoritem izvajal detekcijo in/ali prepoznavo obrazov. Datoteka se ustvari med samim izvajanjem algoritma. Za pravilen format datoteke mora poskrbeti avtor algoritma. Vsebina ustvarjene datoteke se integrira v FDMRL strukturo podatkov ustreznih XML datotek na FAAT strežniku. Integracijo izvede v ta namen implementirani algoritem na strežniku. Algoritem kot vhodni parameter vzame tekstovno datoteko in na podlagi interpretacije vsebine datoteke izvede prepis rezultatov v XML format.

2.5.1 Struktura podatkov

V tem poglavju navajam strukturo tekstovne datoteke (tabela 9), kreirane z algoritmom za detekcijo in/ali prepoznavo obrazov. Pri navajanju podatkov posameznih obrazov ali značilk je možna variabilnost pri samem zapisu koordinat. Tako je, na primer, lahko obraz (ali njegova značilka) definiran z začetno koordinato (podano s točko X in točko Y) ter s širino in višino obraza (ali značilke) ali pa kot poligon štirih koordinat (vsaka koordinata je podana s točko X in točko Y).

Glava datoteke

#Algorithm_URI: vrednost
 #Algorithm_name: vrednost
 #Authors_of_algorithm: vrednost
 #Authors_of_implementation: vrednost
 #Contact_information: vrednost
 #Test_date: vrednost
 #Machine_type: vrednost
 #Implementation_tool: vrednost

Rezultati algoritma

Picture_name1 Detection_time_in_seconds1
 Face1 X Y Width Height Property_name1[Property_value] ...
 Face_feature1 X Y Width Height Property_name1[Property_value] ...
 Face_feature2 X Y Width Height Property_name1[Property_value] ...
 Face_feature3 X Y Width Height Property_name1[Property_value] ...
 ...
 Face2 X Y Width Height Property_name1[Property_value] ...
 Face_feature1 X Y Width Height Property_name1[Property_value] ...
 Face_feature2 X Y Width Height Property_name1[Property_value] ...
 Face_feature3 X Y Width Height Property_name1[Property_value] ...
 ...
 Picture_name2 Detection_time_in_seconds1
 Face1 X Y Width Height Property_name1[Property_value] ...
 Face_feature1 X Y Width Height Property_name1[Property_value] ...
 Face_feature2 X Y Width Height Property_name1[Property_value] ...
 Face_feature3 X Y Width Height Property_name1[Property_value] ...
 ...
 Face2 X Y Width Height Property_name1[Property_value] ...
 Face_feature1 X Y Width Height Property_name1[Property_value] ...
 Face_feature2 X Y Width Height Property_name1[Property_value] ...
 Face_feature3 X Y Width Height Property_name1[Property_value] ...
 ...

Tabela 9: Struktura datoteke z rezultati algoritma.

Priloga C prikazuje format izdelane datoteke z rezultati algoritma.

2.6 Namestitev orodja FAAT

2.6.1 Strežnik

Vse aplikacije orodja FAAT se namestijo na spletni strežnik. Za pravilno delovanje aplikacij je potrebno na strežniku, pred namestitvijo aplikacij, zagotoviti naslednjo programsko opremo:

1. Operacijski sistem tipa Linux,
2. Apache spletni strežnik,
3. Skriptni jezik PHP (verzija 4.0 ali novejši),
4. MySQL podatkovna baza.

Po namestitvi zahtevane programske opreme se izvedejo potrebne namestitve za zagon spletnega strežnika in podatkovne baze. Na spletnem strežniku se ustvari virtualni direktorij (faat), kamor prekopiramo datoteke orodja FAAT. Tabela 9 prikazuje strukturo datotek virtualnega direktorija na spletnem strežniku.

| Direktorijska struktura | Opis |
|-------------------------|--|
| \faat | root direktorij (modul za avtentikacijo) |
| \admin | aplikacija za administracijo sistema |
| \faat_db | aplikacija za prenos baze slik |
| \faat_tool | aplikacija za označevanje slik |
| \help | pomoč uporabnikom |
| \images | baza slik za označevanje |
| \noot_annotated | slike brez označb |
| \needs_approval | slike z označbami, ki čakajo na potrditev |
| \annotated | potrjene slike z označbami |
| \includes | skripte s konstantami, funkcijami, knjižnice |
| \schema | FAAT XML shema |

Tabela 10: Struktura datotek orodja FAAT na spletnem strežniku.

2.6.2 Podatkovna baza

Vse potrebne tabele za pravilno delovanje orodja FAAT generira skripta za namestitev (poglavje 3.6.3). Orodje za delovanje potrebuje dostop do podatkovne baze MySQL. Administrator na strežniku ustvari prazno podatkovno bazo in uporabniški račun za dostop do podatkovne baze.

Konfiguracijska datoteka \faat\faat-config.php vsebuje potrebne konstante za dostop orodja do podatkovne baze. Konstante v tej datoteki (tabela 10) je pred zagonom inštalacijske skripte potrebno nastaviti na pravilne vrednosti:

| Ime konstante | Opis |
|---------------|--|
| DB_NAME | Ime podatkovne baze |
| DB_USER | Uporabniško ime za dostop do podatkovne baze |
| DB_PASSWORD | Uporabniško geslo za dostop do podatkovne baze |
| DB_HOST | Ime strežnika podatkovne baze |

Tabela 11: Tabela konstant za dostop do podatkovne baze MySQL.

2.6.3 Namestitvev

Namestitvena skripta (slike 19 – 21) je zadnji korak namestitve orodja FAAT. Potrebna je za kreiranje tabel v podatkovni bazi in registracijo uporabnika administratorja. Poteka v naslednjih korakih:

1. zagon namestitvene skripte (http://ime_streznika/faat/install.php)



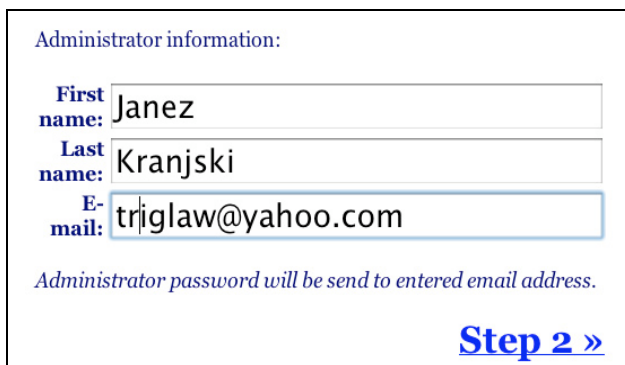
FAAT setup.

- Administrator info ...
- Create tables to FAAT db...
- Finish!

[Step 1 »](#)

Slika 19: Zagon namestitvene skripte.

2. vnos podatkov uporabnika administratorja



Administrator information:

First name:

Last name:

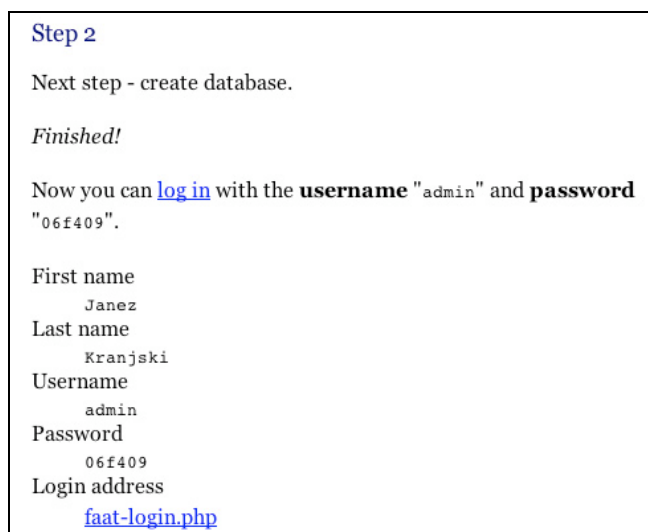
E-mail:

Administrator password will be send to entered email address.

[Step 2 »](#)

Slika 20: Prvi korak namestitve.

3. kreiranje uporabniškega imena in gesla za uporabnika administratorja



Slika 21: Drugi korak namestitve.

3 □ Sklepne ugotovitve

V okviru diplomske naloge smo obravnavali problematiko izdelave orodja za označevanje obrazov na slikah, analize rezultatov algoritmov in izgradnje podatkovne baze slik. Predstavili smo obstoječe podatkovne baze slik in problematiko izgradnje takšne baze. Izpostavili smo pomembnost učinkovitega okolja za razvoj in analizo rezultatov algoritmov ter implementirali orodje za označevanje in analizo človeških obrazov na slikah. Pri pregledu orodij smo potrdili domnevo, da namenskih orodij za označevanje obrazov in integracijo teh označb z rezultati algoritmov ni na voljo. Namesto njih lahko uporabimo nekatera bolj splošna orodja, ki včasih ne zadovoljijo vseh želenih potreb. Predvsem pa je problem ker so posamezne funkcionalnosti implementirane v različnih orodjih, s čimer je s tem njihova integracija oz. integracija izhodnih rezultatov zelo otežena.

V praktičnem delu smo razvili lastno orodje za označevanje in analizo človeških obrazov na slikah. Za razvoj orodja smo naredili temeljito analizo obstoječih orodij in tehnologij ter na podlagi ugotovitev izbrali najprimernejše tehnologije za implementacijo. Pri razvoju smo uporabili princip odjemalec - strežnik. Orodje smo razvili z uporabo spletnih tehnologij PHP, JavaScript, AJAX in XML. Poleg samega orodja smo implementirali lastni jezik za označevanje označenih obrazov na slikah in rezultatov algoritmov, poimenovan FDRML. Shema označevalnega jezika FDRML je objavljena na spletu (dostopno na http://black.fri.uni-lj.si/faat/faat/schema/FDRML_schema.xsd) in je tako na voljo za integracijo v druge podobne sisteme. Pri implementaciji aplikacij smo se držali uporabe odprtokodnih rešitev in neodvisnosti od platforme.

Orodje je bilo nameščeno na testni strežnik in je bilo na voljo omejenemu številu uporabnikov od začetka razvoja. Testiranje orodja je zaradi omejenega časa potekalo vzporedno z razvojem, kar ni najboljša rešitev. Količina neodkritih napak se je nekako zmanjšala s testiranjem več uporabnikov v obdobju razvoja. Zaradi obsežnosti orodja do danes nisem uspel implementirati vseh želenih funkcionalnosti, vendar pa se razvoj na tej točki ni ustavil. Pri razvoju orodja bom sodeloval vsaj do implementacije načrtanih funkcionalnosti, potem pa bom predal razvoj raziskovalni skupini v laboratoriju za računalniški vid.

Glavne probleme pri uporabi orodja vidim v nezadostnem testiranju posameznih funkcionalnosti. Potreben bi bil celoten sistemski in regresivni test posameznih

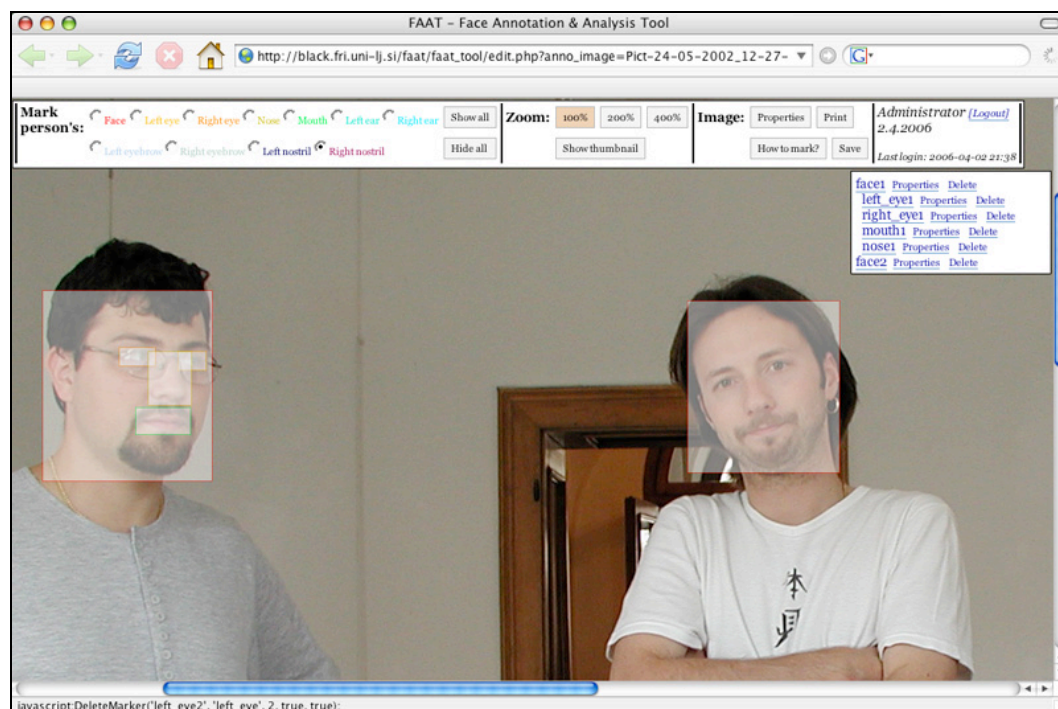
modulov in aplikacij, po potrebi morda tudi stresni test. Druga pomanjkljivost je podpora spletnim odjemalcem. Aplikacija za označevanje človeških obrazov na slikah zaradi različnih implementacij nekaterih tehnologij na strani spletnih odjemalcev ne podpira vseh obstoječih spletnih odjemalcev. Problem je minimiziran, saj aplikacija podpira tri najbolj razširjene spletne odjemalce. Poleg naštetih dveh pomanjkljivosti je možnosti za nadaljnji razvoj in izboljšave še veliko. V nadaljevanju jih navajam le nekaj:

- možnost označevanja poševnih obrazov s poligoni iz štirih točk,
- izboljšanje funkcije zooma in navigacije po sliki,
- implementacija različnih analiz rezultatov algoritmov in kalkulacij,
- implementacija varnostnih mehanizmov ipd.

4 □ Priloge

A. Primer kreiranja XML dokumenta

Primer dokumenta XML in slika uporabniškega vmesnika (slika 22) s podatki označb obrazov, značilk in lastnosti, ki ga orodje ustvari, potem ko uporabnik označi sliko.



Slika 22: Označeni obrazi in značilke v aplikaciji za označevanje obrazov na slikah.

```
<FDMRL xsi:schemaLocation="fdrml http://black.fri.uni-
lj.si/faat/faat/schema/FDRML_schema.xsd" Version="1.0">
  <Image>
    <Image_URI>Pict-24-05-2002_12-27-07.jpg</Image_URI>
    <EXIF_image_width>2048</EXIF_image_width>
    <EXIF_image_height>1536</EXIF_image_height>
    <EXIF_white_balance>0</EXIF_white_balance>
    <EXIF_flash>0</EXIF_flash>
    <EXIF_light_source>String</EXIF_light_source>
    <EXIF_colorspace>1</EXIF_colorspace>
    <EXIF_capture_date_and_time>24:05:2002
12:27:07</EXIF_capture_date_and_time>
    <Ground_truth_author>Borut Batagelj</Ground_truth_author>
    <Ground_truth_date>13.2.2006</Ground_truth_date>
    <Illumination>Daylight</Illumination>
```

```

<Illumination_direction>From infront</Illumination_direction>
<Faces>
  <Face>
    <Left_top_x_coordinate>297</Left_top_x_coordinate>
    <Left_top_y_coordinate>562</Left_top_y_coordinate>
    <Face_width>165</Face_width>
    <Face_height>185</Face_height>
    <Name>facel</Name>
    <Gender>male</Gender>
    <Face_characteristics>glasses, mustaches,
beard</Face_characteristics>
    <Complexion>whyte</Complexion>
    <Face_orientation>Left profile 45° (left
cheek)</Face_orientation>
    <Face_expression>neutral</Face_expression>
    <Age>19-60 years</Age>
    <Hair_color>Black</Hair_color>
    <Hair_length>Short</Hair_length>
    <Probability>1</Probability>
    <Face_features>
      <Face_feature>
        <Left_top_x_coordinate>428</Left_top_x_coordinate>
        <Left_top_y_coordinate>621</Left_top_y_coordinate>
        <Feature_width>28</Feature_width>
        <Feature_height>19</Feature_height>
        <Feature_type>Left eye</Feature_type>
        <Feature_properties>Opened</Feature_properties>
        <Probability>1</Probability>
      </Face_feature>
      <Face_feature>
        <Left_top_x_coordinate>371</Left_top_x_coordinate>
        <Left_top_y_coordinate>617</Left_top_y_coordinate>
        <Feature_width>36</Feature_width>
        <Feature_height>19</Feature_height>
        <Feature_type>Right eye</Feature_type>
        <Feature_properties>Open</Feature_properties>
        <Probability>1</Probability>
      </Face_feature>
      <Face_feature>
        <Left_top_x_coordinate>387</Left_top_x_coordinate>
        <Left_top_y_coordinate>675</Left_top_y_coordinate>
        <Feature_width>54</Feature_width>
        <Feature_height>27</Feature_height>
        <Feature_type>Mouth</Feature_type>
        <Feature_properties>Closed</Feature_properties>
        <Probability>1</Probability>
      </Face_feature>
    </Face_features>
  </Face>

```

```

    <Left_top_x_coordinate>920</Left_top_x_coordinate>
    <Left_top_y_coordinate>572</Left_top_y_coordinate>
    <Face_width>148</Face_width>
    <Face_height>167</Face_height>
    <Name>face2</Name>
    <Gender>male</Gender>
    <Face_characteristics>mustaches,
beard</Face_characteristics>
    <Complexion>whyte</Complexion>
    <Face_orientation>frontal</Face_orientation>
    <Face_expression>neutral</Face_expression>
    <Age>19-60 years</Age>
    <Hair_color>Brown</Hair_color>
    <Probability>1</Probability>
  </Face>
</Faces>
</Image>
</FDMRL>

```

B. XML shema FDRML jezika

Izvorna koda XML sheme jezika FDRML za označevanje obrazov, značilk in njihovih lastnosti. Shema je javno objavljena na spletu in je dostopna na naslovu http://black.fri.uni-lj.si/faat/faat/schema/FDRML_schema.xsd.

```
<?xml version="1.0" encoding="UTF-8" ?>
<xs:schema xmlns="fdrml"
xmlns:xs="http://www.w3.org/2001/XMLSchema" targetNamespace="fdrml"
elementFormDefault="qualified" attributeFormDefault="unqualified">
<xs:element name="FDRML">
<xs:annotation>
  <xs:documentation>Initial tag on which a data file is
built</xs:documentation>
</xs:annotation>
<xs:complexType>
<xs:sequence>
  <xs:element ref="Image" />
</xs:sequence>
<xs:attribute name="Version" type="xs:string" use="required">
<xs:annotation>
  <xs:documentation>Version of the XML Schema</xs:documentation>
</xs:annotation>
</xs:attribute>
</xs:complexType>
</xs:element>
<xs:element name="Image">
<xs:annotation>
  <xs:documentation>Image description tag</xs:documentation>
</xs:annotation>
<xs:complexType>
<xs:sequence>
<xs:element name="Image_URI" type="xs:string">
<xs:annotation>
  <xs:documentation>Location (Universal Resource Identifier) where
the annotation image can be found</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element name="EXIF_image_width" type="xs:int" />
<xs:element name="EXIF_image_height" type="xs:int" />
<xs:element name="EXIF_white_balance" type="xs:string" />
<xs:element name="EXIF_flash" type="xs:string" />
<xs:element name="EXIF_light_source" type="xs:string" />
<xs:element name="EXIF_colorspace" type="xs:string" />
<xs:element name="EXIF_capture_date_and_time" type="xs:dateTime"
/>
/>
```

```

<xs:element name="Ground_truth_author" type="xs:string">
  <xs:annotation>
    <xs:documentation>Author of image annotation</xs:documentation>
  </xs:annotation>
</xs:element>
<xs:element name="Ground_truth_date" type="xs:date">
  <xs:annotation>
    <xs:documentation>Date of image annotation</xs:documentation>
  </xs:annotation>
</xs:element>
<xs:element name="Illumination" type="xs:string">
  <xs:annotation>
    <xs:documentation>Type of illumination used when picture was
taken</xs:documentation>
  </xs:annotation>
</xs:element>
<xs:element name="Illumination_direction" type="xs:string"
minOccurs="0">
  <xs:annotation>
    <xs:documentation>Direction of that
illumination</xs:documentation>
  </xs:annotation>
</xs:element>
<xs:element name="Background" type="xs:string" minOccurs="0">
  <xs:annotation>
    <xs:documentation>Image background type</xs:documentation>
  </xs:annotation>
</xs:element>
<xs:element name="Comment" type="xs:string" minOccurs="0">
  <xs:annotation>
    <xs:documentation>Additional details about the
image</xs:documentation>
  </xs:annotation>
</xs:element>
<xs:element ref="Faces" minOccurs="0" />
<xs:element ref="Face_detection_algorithms" minOccurs="0" />
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="Face_detection_algorithms">
  <xs:annotation>
    <xs:documentation>Summary tag for all computer vision algorithms
applied to this image</xs:documentation>
  </xs:annotation>
<xs:complexType>
  <xs:sequence>
    <xs:element ref="Face_detection_algorithm" maxOccurs="unbounded"
/>
  </xs:sequence>

```

```

    </xs:complexType>
  </xs:element>
<xs:element name="Face_detection_algorithm">
<xs:annotation>
  <xs:documentation>Description of a single computer vision
algorithm
applied to this image</xs:documentation>
  </xs:annotation>
<xs:complexType>
<xs:sequence>
<xs:element name="Algorithm_URI" type="xs:string">
<xs:annotation>
  <xs:documentation>Location (Universal Resource Identifier) where
the
algorithm can be found</xs:documentation>
  </xs:annotation>
  </xs:element>
<xs:element name="Algorithm_name" type="xs:string">
<xs:annotation>
  <xs:documentation>Logical name of the
algorithm</xs:documentation>
  </xs:annotation>
  </xs:element>
<xs:element name="Authors_of_algorithm" type="xs:string">
<xs:annotation>
  <xs:documentation>Authors of this algorithm</xs:documentation>
  </xs:annotation>
  </xs:element>
<xs:element name="Authors_of_implementation" type="xs:string">
<xs:annotation>
  <xs:documentation>Authors of algorithm
implementation</xs:documentation>
  </xs:annotation>
  </xs:element>
<xs:element name="Contact_information" type="xs:string">
<xs:annotation>
  <xs:documentation>Authors contact information</xs:documentation>
  </xs:annotation>
  </xs:element>
<xs:element name="Test_date" type="xs:date">
<xs:annotation>
  <xs:documentation>Date on which the test was
performed</xs:documentation>
  </xs:annotation>
  </xs:element>
<xs:element name="Machine_type" type="xs:string">
<xs:annotation>
  <xs:documentation>Description of the computer used for
testing</xs:documentation>

```

```

    </xs:annotation>
  </xs:element>
<xs:element name="Implementation_tool" type="xs:string">
<xs:annotation>
  <xs:documentation>Tool used to implement this
algorithm</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element name="Detection_time" type="xs:duration">
<xs:annotation>
  <xs:documentation>Time needed to detect faces in the
image</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element name="Comment" type="xs:string" minOccurs="0">
<xs:annotation>
  <xs:documentation>Additional details about the
algorithm</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element ref="Faces" minOccurs="0" />
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="Faces">
<xs:annotation>
  <xs:documentation>Summary tag for all faces on this
image</xs:documentation>
</xs:annotation>
<xs:complexType>
<xs:sequence>
  <xs:element ref="Face" minOccurs="0" maxOccurs="unbounded" />
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="Face">
<xs:annotation>
  <xs:documentation>Description of a single face on the
image</xs:documentation>
</xs:annotation>
<xs:complexType>
<xs:sequence>
<xs:element name="Left_top_x_coordinate" type="xs:int">
<xs:annotation>
  <xs:documentation>Top x coordinate on the image of this
face</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element name="Left_top_y_coordinate" type="xs:int">

```

```

<xs:annotation>
  <xs:documentation>Top y coordinate on the image of this
face</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element name="Face_width" type="xs:int" />
<xs:element name="Face_height" type="xs:int" />
<xs:element name="Name" type="xs:string" minOccurs="0">
<xs:annotation>
  <xs:documentation>Name of this face</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element name="Gender" type="xs:string" minOccurs="0">
<xs:annotation>
  <xs:documentation>Person's gender</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element name="Face_characteristics" type="xs:string"
minOccurs="0" />
<xs:element name="Complexion" type="xs:string" minOccurs="0">
<xs:annotation>
  <xs:documentation>Complexion of this face, defining person's
rase</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element name="Face_orientation" type="xs:string" minOccurs="0">
<xs:annotation>
  <xs:documentation>Defines orientation of this face towards the
camera</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element name="Face_expression" type="xs:string" minOccurs="0"
/>
<xs:element name="Age" type="xs:short" minOccurs="0" />
<xs:element name="Hair_color" type="xs:string" minOccurs="0" />
<xs:element name="Hair_length" type="xs:string" minOccurs="0" />
<xs:element name="Face_skin" type="xs:string" minOccurs="0" />
<xs:element name="Comment" type="xs:string" minOccurs="0">
<xs:annotation>
  <xs:documentation>Additional details about the
image</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element name="Probability" type="xs:string" minOccurs="0">
<xs:annotation>
  <xs:documentation>Probability that this face is a true
face</xs:documentation>
</xs:annotation>
</xs:element>

```



```

    <xs:element ref="Face_features" minOccurs="0" />
  </xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="Face_features">
  <xs:annotation>
    <xs:documentation>Summary tag for all face features on a
face</xs:documentation>
  </xs:annotation>
<xs:complexType>
  <xs:sequence>
    <xs:element ref="Face_feature" maxOccurs="11" />
  </xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="Face_feature">
  <xs:annotation>
    <xs:documentation>Description of a single face
feature</xs:documentation>
  </xs:annotation>
<xs:complexType>
  <xs:sequence>
    <xs:element name="Left_top_x_coordinate" type="xs:int">
      <xs:annotation>
        <xs:documentation>Top x coordinate on the image of this face
feature</xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element name="Left_top_y_coordinate" type="xs:int">
      <xs:annotation>
        <xs:documentation>Top y coordinate on the image of this face
feature</xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element name="Feature_width" type="xs:int" />
    <xs:element name="Feature_height" type="xs:int" />
    <xs:element name="Feature_type" type="xs:string">
      <xs:annotation>
        <xs:documentation>Defining type of face
feature</xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element name="Feature_properties" type="xs:string"
minOccurs="0">
      <xs:annotation>
        <xs:documentation>List of properties of this face
fetaure</xs:documentation>
      </xs:annotation>
    </xs:element>

```

```
<xs:element name="Comment" type="xs:string" minOccurs="0">
  <xs:annotation>
    <xs:documentation>Additional details about theis face
feature</xs:documentation>
  </xs:annotation>
</xs:element>
<xs:element name="Probability" type="xs:string" minOccurs="0">
  <xs:annotation>
    <xs:documentation>Probability that this face feature is a true
face_feature</xs:documentation>
  </xs:annotation>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:schema>
```

C. Datoteka z rezultati algoritma

Primer tekstovne datoteke z rezultati algoritma. Prikazani so rezultati algoritma, ki na sliki detektira obraze.

```
#Algorithm_URI
http://research.microsoft.com/viola/Pubs/Detect/violaJones_IJCV.pdf
#Algorithm_name: AdaBoost&Haar
#Authors_of_algorithm P. Vola and M. Jones
#Authors_of_implementation Intel OpenCV +
haarcascade_frontalface_alt.xml
#Contact_information borut.batagelj@fri.uni-lj.si
#Test_date 15.3.2006
#Machine_type AMD Athlon 64 X2 Dual, 4200+, 2.42 GHz, 2.00 GB RAM
#implementation_tool= C++
#Detection time

1\MVC-001F.JPG
1\MVC-002F.JPG 1.12 s
face 274 152 171 171
1\MVC-003F.JPG 0.88 s
face 240 148 171 171
1\MVC-004F.JPG 0.85 s
face 192 155 132 132
1\MVC-005F.JPG
1\MVC-006F.JPG 0.83 s
face 222 180 166 166
1\MVC-007F.JPG 0.83 s
face 225 177 164 164
10\MVC-001F.JPG
10\MVC-002F.JPG
10\MVC-003F.JPG 0.81 s
face 203 155 239 239
10\MVC-004F.JPG
10\MVC-005F.JPG 0.85 s
face 194 295 75 75
10\MVC-006F.JPG 0.89 s
face 184 148 264 264
10\MVC-007F.JPG 0.83 s
face 193 173 245 245
100\MVC-001F.JPG
100\MVC-002F.JPG
100\MVC-003F.JPG 0.76 s
face 483 413 32 32
face 210 130 212 212
100\MVC-004F.JPG
```

100\MVC-005F.JPG
100\MVC-006F.JPG 0.82 s
face 492 413 33 33
face 196 117 232 232
100\MVC-007F.JPG 0.80 s
face 448 381 35 35
face 166 120 234 234
101\MVC-001F.JPG
101\MVC-002F.JPG 0.76 s
face 334 217 91 91
101\MVC-003F.JPG 0.70 s
face 217 127 222 222
101\MVC-004F.JPG
101\MVC-005F.JPG
101\MVC-006F.JPG 0.70 s
face 183 121 215 215
101\MVC-007F.JPG 0.73 s
face 190 112 213 213
102\MVC-001F.JPG
102\MVC-002F.JPG
102\MVC-003F.JPG 0.81 s
face 214 125 235 235
102\MVC-004F.JPG 0.78 s
face 185 162 155 155
102\MVC-005F.JPG 0.76 s
face 169 175 78 78
102\MVC-006F.JPG 0.83 s
face 219 131 206 206
102\MVC-007F.JPG 0.86 s
face 211 128 211 211
103\MVC-001F.JPG
103\MVC-002F.JPG
103\MVC-003F.JPG 0.81 s
face 168 120 206 206

Spisek slik

| | |
|--|----|
| SLIKA 1: ALGORITEM SISTEMA ZA ISKANJE IN RAZPOZNAVANJE OBRAZOV..... | 4 |
| SLIKA 2: VARIACIJE POZ V BAZI CVL..... | 13 |
| SLIKA 3: SHEMA ORODJA LABELME..... | 15 |
| SLIKA 4: KLASIČNI MODEL SPLETNE STRANI (LEVO) IN AJAX MODEL SPLETNE STRANI (DESNO)..... | 19 |
| SLIKA 5: RAZLIKA MED RAZLIČNIMA MODELOMA SPLETNIH STRANI..... | 20 |
| SLIKA 6: UPORABNIŠKI MODEL ZA SCENARIJ 1 – UPORABNIK ANOTACIJE..... | 22 |
| SLIKA 7: UPORABNIŠKI MODEL ZA SCENARIJ 2 – ADMINISTRATOR..... | 23 |
| SLIKA 8: UPORABNIŠKI MODEL ZA SCENARIJ 3 – UPORABNIK PODATKOVNE BAZE..... | 24 |
| SLIKA 9: UPORABNIŠKI MODEL ZA SCENARIJ 4 – UPORABNIK ANALIZE..... | 24 |
| SLIKA 10: ARHITEKTURA ORODJA FAAT..... | 26 |
| SLIKA 11: SHEMA UPORABNIŠKIH PRAVIC SISTEMA..... | 27 |
| SLIKA 12: AVTENTIKACIJA UPORABNIKOV..... | 28 |
| SLIKA 13: REGISTRACIJA UPORABNIKOV..... | 29 |
| SLIKA 14: UPORABNIŠKI VMESNIK APLIKACIJE ZA OZNAČEVANJE..... | 30 |
| SLIKA 15: SHEMA LASTNOSTI OBRAZA..... | 31 |
| SLIKA 16: SHEMA LASTNOSTI OBRAZNIH ZNAČILK..... | 32 |
| SLIKA 17: APLIKACIJA ZA PRENOS BAZE SLIK..... | 33 |
| SLIKA 18: MODEL STRUKTURE PODATKOV JEZIKA FDRML..... | 36 |
| SLIKA 19: ZAGON NAMESTITVENE SKRIPTE..... | 40 |
| SLIKA 20: PRVI KORAK NAMESTITVE..... | 40 |
| SLIKA 21: DRUGI KORAK NAMESTITVE..... | 41 |
| SLIKA 22: OZNAČENI OBRAZI IN ZNAČILKE V APLIKACIJI ZA OZNAČEVANJE OBRAZOV NA SLIKAH. ... | 44 |

Spisek tabel

| | |
|--|----|
| TABELA 1: STATISTIKA PODATKOVNE BAZE OBRAZOV AR. | 9 |
| TABELA 2: STATISTIKA PODATKOVNE BAZE OBRAZOV BANCA. | 10 |
| TABELA 3: STATISTIKA PODATKOVNE BAZE OBRAZOV CAS-PEAL. | 10 |
| TABELA 4: STATISTIKA PODATKOVNE BAZE OBRAZOV CMU - HYPERSPECTRAL. | 11 |
| TABELA 5: STATISTIKA PODATKOVNE BAZE OBRAZOV CMU PIE. | 11 |
| TABELA 6: STATISTIKA PODATKOVNE BAZE OBRAZOV FERET. | 11 |
| TABELA 7: STATISTIKA KOREJSKE PODATKOVNE BAZE OBRAZOV (KFDB). | 12 |
| TABELA 8: STATISTIKA PODATKOVNE BAZE OBRAZOV CVL. | 12 |
| TABELA 9: STRUKTURA DATOTEKE Z REZULTATI ALGORITMA. | 38 |
| TABELA 10: STRUKTURA DATOTEK ORODJA FAAT NA SPLETNEM STREŽNIKU. | 39 |
| TABELA 11: TABELA KONSTANT ZA DOSTOP DO PODATKOVNE BAZE MySQL. | 40 |

5 □ Literatura in viri

- [1] R. Gross, Handbook of Face Recognition, poglavje »Face databases«, S.Li, A.Jain, ed., Springer, New York, February, 2005.
- [2] P. Peer, Avtomatsko iskanje človeških obrazov na slikah, diplomska naloga na Fakulteti za računalništvo in informatiko, Univerza v Ljubljani, sept. 1998.
- [3] B. Batagelj, F. Solina, Uporaba sistemov za prepoznavo človeških obrazov, članek objavljen na konferenci Rosus, Rosus, 2006.
- [4] F. Solina, Računalniški vid nekoč in danes, članek objavljen na konferenci Rosus, Rosus, 2006.
- [5] F. Solina, P. Peer, B. Batagelj, S. Juvan, J. Kovač, Color-Based Face Detection in the »15 Seconds of Fame« Art Installation, International Conference on Computer Vision / Computer Graphics Collaboration for Model-based Imaging, Rendering, image Analysis and Graphical special Effects MIRAGE'03, Ed. W. Philips, pp. 38-47, INRIA Rocquencourt, France, March 2003.
- [6] J. J. Gareth, Ajax: A New Approach to Web Applications, Adaptive Path Publications. Dostopno na <http://adaptivepath.com/publications/essays/archives/000385.php>.
- [7] B. Russell, A. Torralba, W. T. Freeman, LabelME Open Annotation Tool, Computer Science and Artificial Intelligence Laboratory at MIT, 2005. Dostopno na <http://people.csail.mit.edu/brussell/research/LabelMe/intro.html>.
- [8] (2006) Face Detection Homepage. Dostopno na <http://www.facedetection.com>.
- [9] (2006) Spletna enciklopedija Wikipedia. Dostopno na www.wikipedia.com.
- [10] (2006) World Wide Web Consortium. Dostopno na www.w3.org.

Izjava o avtorstvu

Izjavljam, da sem diplomsko delo izdelal samostojno pod vodstvom mentorja (Orodje za označevanje in analizo človeških obrazov na slikah, prof. dr. Franc Solina). Izkazano pomoč drugih sodelavcev sem v celoti navedel v zahvali.

V Ljubljani, 15. 4. 2006

Primož Rome